Simplon getting started

Introduction and quick start guide

Leutron Vision

Simplon getting started: Introduction and quick start guide

Leutron Vision

Revision: 1.0-17273. Leutron Vision documentation set.

Publication date 28 April 2011

Copyright © 1995-2011 Leutron Vision

All Information in this document is subject to change without notice and does not represent a commitment on the part of Leutron Vision. The software products described in this document are furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of agreement.

It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement. The licensee may make one copy of the software for backup purposes. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the licensee's personal use, without the express written permission of Leutron Vision.

Product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Table of Contents

General information

Scope of the manual

This manual walks you through the Simplon software package installation process on various operating systems, suggests how to test the installation and provides information about various configuration and troubleshooting options for Simplon and its components. It also provides high-level overview of the package and its relationship with industry standards.

Related documents

- Simplon programmer's guide — overview and tutorial of the Simplon API library.
- PicSight-GigE user manual — user manual for our GigE Vision camera series.
- CheckSight user manual — user manual for our CheckSight smart camera series.
- Simplon tools — overview and documentation of various tools delivered with the Simplon package.

1. Quick start

This chapter provides instructions how to quickly set up and test a system using the Simplon package with Leutron Vision hardware. Individual sections discuss following tasks:

- Connecting the camera — install, connect and configure the cameras
- Installation — quick quide through Simplon installation process under Windows
- Basic camera configuration — prepare the camera for the first acquisition
- Test the camera — instructions how to verify camera functionality in Simplon Explorer
- Further directions — links to other important information

1.1. Connecting and configuring the camera(s)

Before installing the software and testing the functionality, let's prepare the camera(s) for use.

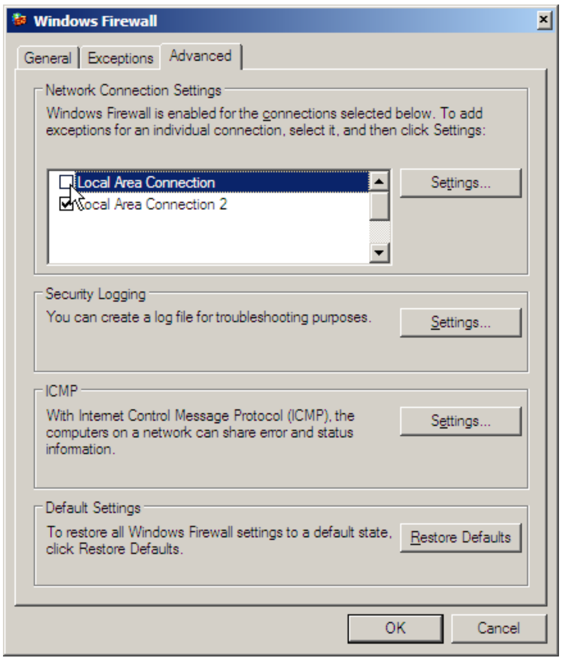
The range of Leutron Vision camera selection consists of several main families:

- PicSight-GigE — GigE Vision cameras (Gigabit Ethernet interface); hardware manual: PicSight-GigE user manual
- PicSight-Smart — advanced version of PicSight-GigE, supporting development on an embedded processor; hardware manual: PicSight-GigE user manual
- CheckSight — true PC based smart camera; hardware manual: CheckSight user manual

All camera families share the same full choice of image sensors, from VGA resolution to several megapixels, CCD and CMOS, various sensor and pixel sizes, functionality and price range.

1.1.1. PicSight-GigE & PicSight-Smart

1. Consult precautions listed in PicSight-GigE user manual. Be sure to not violate instructions in the manual.
2. Connect the camera to proper 10—26V power supply.
3. Connect the camera to the network, or directly to the PC using a suitable Ethernet cable. We recommend to use shielded (S/STP) category 6 cable or better.
4. If you have a DHCP server running in the network, the camera will get a proper IP address automatically. Otherwise you will need to adjust the IP configuration of the camera using tools from the Simplon package, once it is installed (other software packages might also offer IP configuration tools). We'll come back to this later.
5. In optimal case, the network card interfacing the camera will be dedicated to the image acquisition only and thus will have no access to internet. In such case the firewall can be fully disabled for this network interface. In Windows Firewall (configured in Control Panel) this can be selected in the Advanced configuration



Network Connection Settings

Windows Firewall is enabled for the connections selected below. To add exceptions for an individual connection, select it, and then click Settings:

□ Local Area Connection
☑ Local Area Connection 2

Security Logging

You can create a log file for troubleshooting purposes.

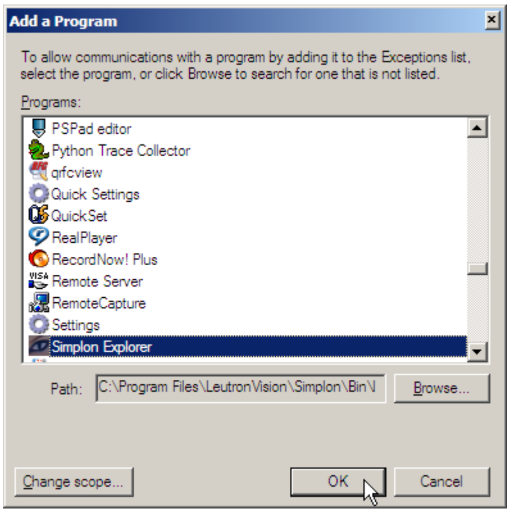
ICMP

With Internet Control Message Protocol (ICMP), the computers on a network can share error and status information.

Default Settings

To restore all Windows Firewall settings to a default state, click Restore Defaults.

6. If your network configuration, however, requires to share internet and other network access with the image acquisition over the same media (which is not recommended practice), you will need to configure your firewall so that an it does not block the communication with the camera. In Windows Firewall this can be done by creating an exception for each application that will be accessing the camera — especially for the Simplon Explorer that we will use for the first tests (possibly limiting the exception's scope to the local subnet only). Note that the applications to be unblocked (including Simplon Explorer) are not usually present yet in the system at the time of installation, therefore this step might have to be repeated after the installation itself and again for each new application. Therefore we again recommend to have a network connection dedicated solely for camera connection — and disabling firewall fully for this connection.

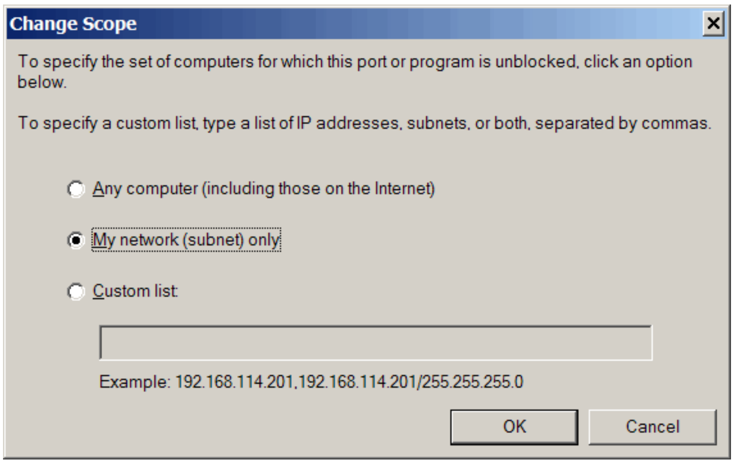


To allow communications with a program by adding it to the Exceptions list, select the program, or click Browse to search for one that is not listed.

Programs:

PSPad editor
Python Trace Collector
qrfcview
Quick Settings
QuickSet
RealPlayer
RecordNow! Plus
Remote Server
RemoteCapture
Settings
Simplon Explorer

Path: C:\Program Files\LeutronVision\Simplon\Bin\I



Change Scope

To specify the set of computers for which this port or program is unblocked, click an option below.

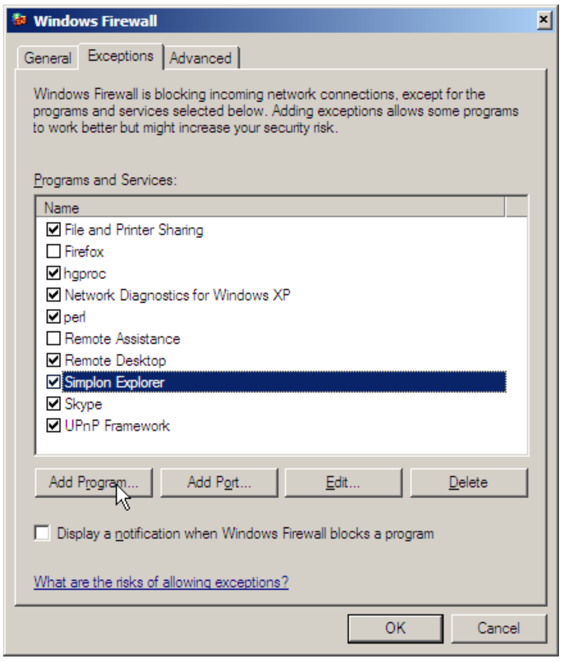
To specify a custom list, type a list of IP addresses, subnets, or both, separated by commas.

○ Any computer (including those on the Internet)

● My network (subnet) only

○ Custom list:

Example: 192.168.114.201,192.168.114.201/255.255.255.0



Windows Firewall

Windows Firewall is blocking incoming network connections, except for the programs and services selected below. Adding exceptions allows some programs to work better but might increase your security risk.

Programs and Services:

Name
☑ File and Printer Sharing
☐ Firefox
☑ hgproc
☑ Network Diagnostics for Windows XP
☑ perl
☐ Remote Assistance
☑ Remote Desktop
☑ Simplon Explorer
☑ Skype
☑ UPnP Framework

☐ Display a notification when Windows Firewall blocks a program

What are the risks of allowing exceptions?

1.1.2. CheckSight

1. Consult precautions listed in CheckSight user manual. Be sure to not violate instructions in the manual.
2. Unpack the camera, connect necessary peripherals and power it up.
3. Assuming that the desired operating system is already pre-installed on the camera, no additional steps are needed and the camera is ready for Simplon installation ant test.

1.2. Simplon installation — quick guide for Windows

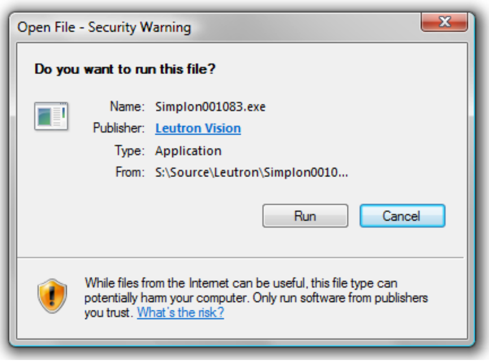
Following points guide you quickly through Simplon installation under Windows. If in doubts, if you need detailed instructions or if installing in another operating system, refer to full Simplon installation description [p. 40].

Now let's walk through the installation procedure:

1. Prepare the installation media. The software should be installed by a user with administrator rights.

- If you have the installation CD, insert it in the CD-ROM drive and wait until installation starts.
- If you have downloaded the installer from our website[1], execute it.

2. Depending on security adjustments of your system and whether you are running the installer from a local disk or a network share, you might get following security warning. Simply proceed clicking Run.

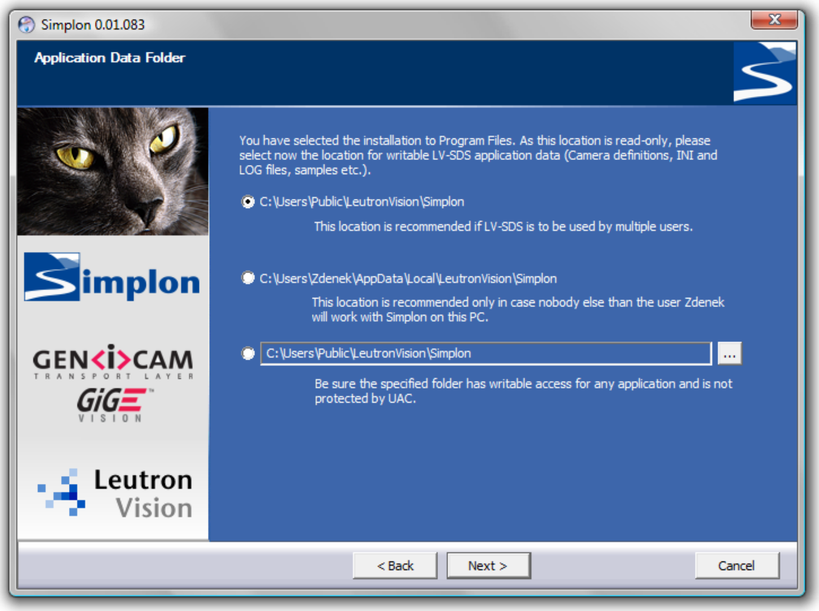

Open File - Security Warning

Do you want to run this file?

Name: Simplon001083.exe
Publisher: Leutron Vision
Type: Application
From: S:\Source\Leutron\Simplon0010...

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

3. The installer welcomes you with following screen, click Next.
4. Next screen is the license agreement. Please read carefully the license text and if you agree, click the I Agree button to proceed with installation.
5. Next the installer asks you about the location where to install Simplon. The default location will do well in most cases. Keep it or select another preferred location and click Next.
6. If running Windows Vista or newer Windows version, the default software installation directory (`C:\Program Files`) is read only. The installer will ask you where to put the application data requiring write access (the dialog will not appear on Windows XP). Select a desired location and click Next.


Simplon 0.01.083

Application Data Folder

You have selected the installation to Program Files. As this location is read-only, please select now the location for writable LV-SDS application data (Camera definitions, INI and LOG files, samples etc.).

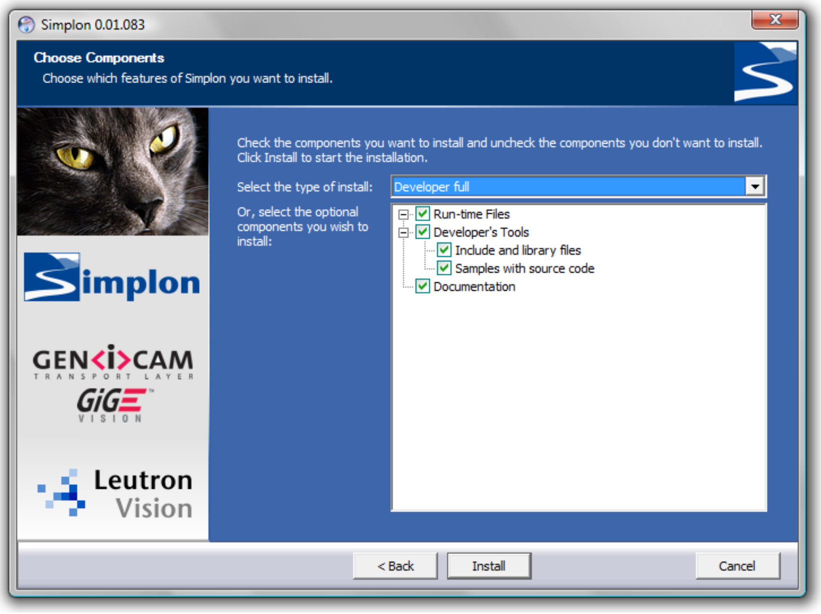
● C:\Users\Public\LeutronVision\Simplon
This location is recommended if LV-SDS is to be used by multiple users.

○ C:\Users\Zdenek\AppData\Local\LeutronVision\Simplon
This location is recommended only in case nobody else than the user Zdenek will work with Simplon on this PC.

○ C:\Users\Public\LeutronVision\Simplon
Be sure the specified folder has writable access for any application and is not protected by UAC.

[1] http://www.leutron.com/support-downloads/download-area/

7. In the next step you can select the Simplon components to install. Note that for testing the cameras, the “runtime” components suffice. However, for programming with Simplon you need to make sure the Developer's tools are installed as well. When finished with the selection, click Install.



Simplon 0.01.083

Choose Components

Choose which features of Simplon you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.

Select the type of install: Developer full

Or, select the optional components you wish to install:

- ☑ Run-time Files
- ☑ Developer's Tools
  - ☑ Include and library files
  - ☑ Samples with source code
- ☑ Documentation

8. After starting the installation, the selected components will be installed to your computer. The installer will inform you about the progress.

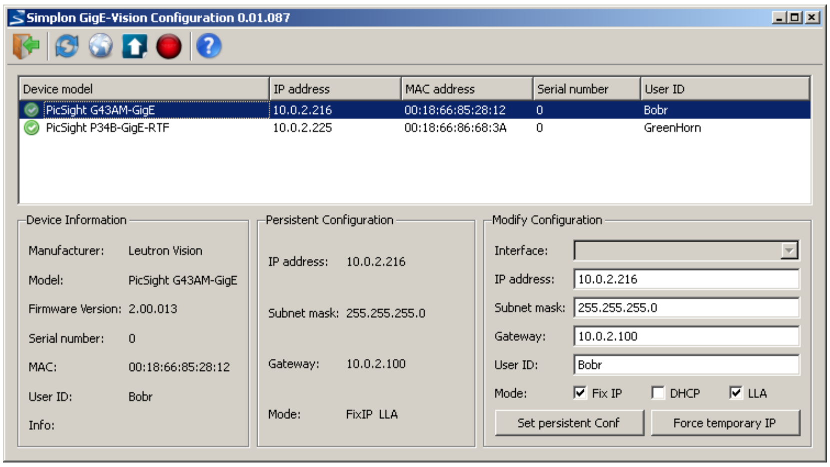
9. When completed, the installer displays the final dialog. Keep the default option (Reboot now) and click Finish. After rebooting, Simplon is ready to use.

1.3. Basic camera configuration

For CheckSight cameras no additional actions are needed, you can directly proceed with testing the camera.

When using PicSight-GigE/PicSight-Smart cameras, you might need to adjust their IP configuration so that it fits your network settings.

Start the Simplon GigE Vision Configuration Tool: Start → Programs → Leutron Vision Simplon → Simplon GigE Vision Configuration Tool (under Linux, start `/opt/simplon/bin/lv.ipconf`). The tool will scan for the GigE Vision cameras connected to the system. If the camera(s) is directly accessible from your PC, they will be displayed with a green “OK” icon ✓. In such case the camera(s) is ready for acquisition, no further actions are necessary.



Simplon GigE-Vision Configuration 0.01.087

Device model	IP address	MAC address	Serial number	User ID
PicSight G43AM-GigE	10.0.2.216	00:18:66:85:28:12	0	Bobr
PicSight P34B-GigE-RTF	10.0.2.225	00:18:66:86:68:3A	0	GreenHorn

Device Information

Manufacturer: Leutron Vision
Model: PicSight G43AM-GigE
Firmware Version: 2.00.013
Serial number: 0
MAC: 00:18:66:85:28:12
User ID: Bobr
Info:

Persistent Configuration

IP address: 10.0.2.216
Subnet mask: 255.255.255.0
Gateway: 10.0.2.100
Mode: FixIP LLA

Modify Configuration

Interface:
IP address: 10.0.2.216
Subnet mask: 255.255.255.0
Gateway: 10.0.2.100
User ID: Bobr
Mode: ☑ Fix IP ☐ DHCP ☑ LLA

If the camera is displayed using a yellow icon ⚠, it is “visible” from your PC, but cannot be directly accessed, because it is configured with different subnet than your network card. In such case, select the camera in the list, adjust its IP configuration so that it matches settings of your network card, select the Fix IP checkbox and click the Set persistent Conf button. When finished, the camera's IP configuration should match the network card. It will display with the green icon in the list and will be now directly accessible from the PC. It is ready for test.

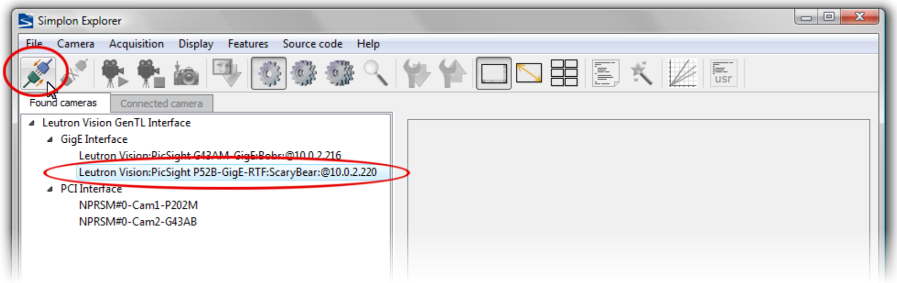
If desired, the camera's “nickname” (user name) can be adjusted as well, together with the IP configuration. The nickname can be used to identify individual cameras connected to the network.

1.4. Testing the camera(s)

The Simplon package contains the Simplon Explorer tool, which allows to enumerate, connect and configure camera(s), acquire images or generate sample source code for Simplon API. It is a useful tool for testing both the Simplon and camera functionality.

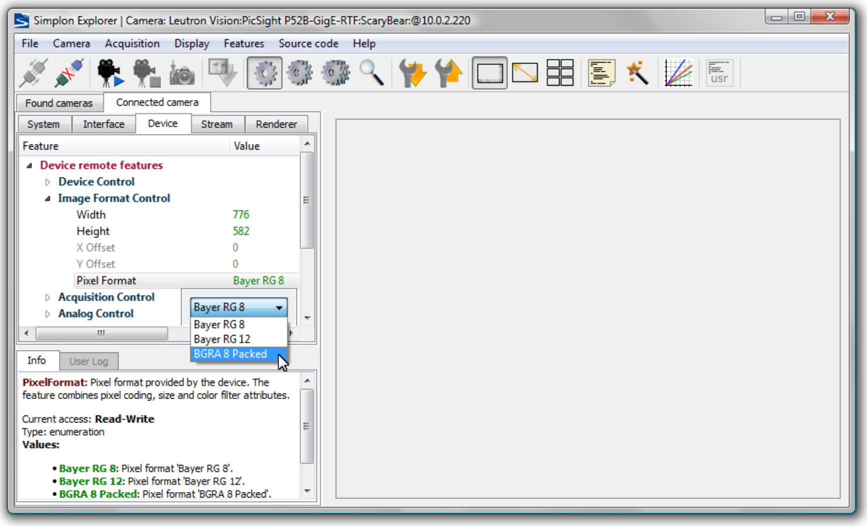
1. Start the tool from system menu: Start → Programs → Leutron Vision Simplon → Simplon Explorer (under Linux, start `/opt/simplon/bin/lv.explorer`).

2. Connect the camera to be tested — select it in the list of found cameras and press the *Connect camera* button. Of course, the camera must be connected and powered.



Leutron Vision GenTL Interface
GigE Interface
Leutron Vision:PicSight G43AM-GigE:Bobr:@10.0.2.216
Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
PCI Interface
NPRSM#0-Cam1-P202M
NPRSM#0-Cam2-G43AB

3. Configure the camera features as desired. Pay attention especially to features in `Image Format Control` and `Acquisition Control` categories.


Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
File Camera Acquisition Display Features Source code Help
usr
Found cameras
Connected camera
System
Interface
Device
Stream
Renderer
Info
User Log

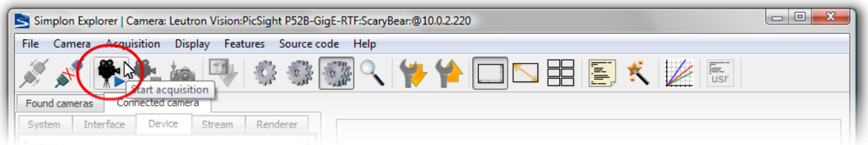
Feature	Value
Device remote features	
Device Control	
Image Format Control	
Width	776
Height	582
X Offset	0
Y Offset	0
Pixel Format	Bayer RG 8
Acquisition Control	Bayer RG 8 Bayer RG 8 Bayer RG 12 BGRA 8 Packed
Analog Control	

PixelFormat: Pixel format provided by the device. The feature combines pixel coding, size and color filter attributes.

Current access: **Read-Write**
Type: enumeration
Values:

- **Bayer RG 8:** Pixel format 'Bayer RG 8'.
- **Bayer RG 12:** Pixel format 'Bayer RG 12'.
- **BGRA 8 Packed:** Pixel format 'BGRA 8 Packed'.

4. When prepared, click the Start acquisition button — the camera should start acquiring. If not, verify again, if the camera is properly connected and running, if the system was properly configured (remember the notes above about firewall, network configuration, etc.) or if the camera was not set to triggered mode, while no trigger was attached.


Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
File Camera Acquisition Display Features Source code Help
Start acquisition
usr
Found cameras
Connected camera
System
Interface
Device
Stream
Renderer

5. While the acquisition is active, you can still adjust the runtime parameters, such as exposure time or gain. Basic acquisition parameters, such as pixel format or trigger mode become locked when the acquisition starts.

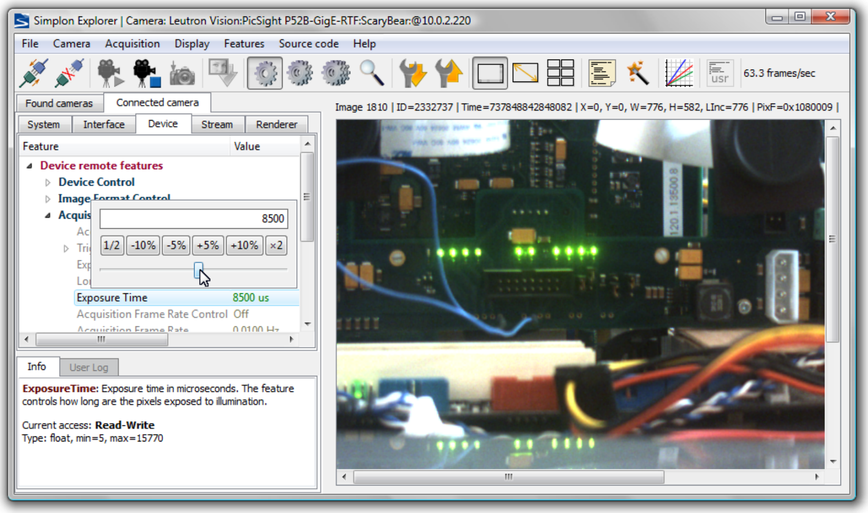

Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
File Camera Acquisition Display Features Source code Help
usr
63.3 frames/sec
Found cameras
Connected camera
System
Interface
Device
Stream
Renderer
8500
1/2
-10%
-5%
+5%
+10%
x2
Info
User Log

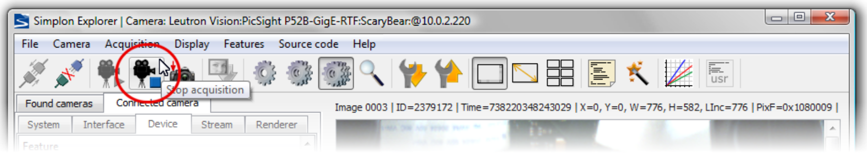
Image 1810 | ID=2332737 | Time=737848842848082 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0x1080009 |

Feature	Value
Device remote features	
Device Control	
Image Format Control	
Acquis	
Exposure Time	8500 us
Acquisition Frame Rate Control	Off
Acquisition Frame Rate	0.0100 Hz

ExposureTime: Exposure time in microseconds. The feature controls how long are the pixels exposed to illumination.

Current access: **Read-Write**
Type: float, min=5, max=15770

6. When finished, click the Stop acquisition button and exit.



Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220

Image 0003 | ID=2379172 | Time=738220348243029 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0x1080009 |

1.5. Where to go next?

Once finished with the basic tests and verification, you might want to:

- Get more detailed information about installation process in all supported operating systems (Chapter 5, *Installation details* [p. 40]) or information about customizing the standard installer (Chapter 6, *Custom installation* [p. 55]).

- Learn about Simplon configuration options (Section 7.1, “Simplon Settings utility and the configuration file” [p. 65]) and overview of troubleshooting and technical support approach (Chapter 9, *Troubleshooting & Support* [p. 70]).

- Get overview about Simplon package, its components, functionality and connectivity (Chapter 2, *Simplon overview* [p. 15]), including the Simplon Explorer tool (Chapter 3, *Simplon Explorer in detail* [p. 22]). Learn to understand the GenICam based camera feature tree and other important Simplon API principles (Chapter 4, *Important principles behind Simplon interface* [p. 34]).

- Know basics about the important machine vision standards, GenICam and GigE Vision (Chapter 8, *Vision Standards* [p. 67]) and how you can take advantage of advanced support for these standards in Simplon when interfacing 3rd party software and hardware products.

- Study documentation for your hardware:
  - PicSight-GigE user manual
  - CheckSight user manual

- Study documentation for the Simplon API:
  - Simplon programmer's guide
  - Simplon reference

2. Simplon overview

This chapter gives a high level descritpion of the Simplon package, its components and the supported hardware.

2.1. Connecting your application with our cameras

The Simplon package is available in versions for Windows and Linux operating systems, running on 32-bit and 64-bit PC architectures.

All supported platforms are since beginning treated with the same priority, share the equivalent feature range and provide the same toolset. Releases for all platforms are synchronized and all tests are executed on every platform. Migrating to a different operating system or upgrading to a 64-bit architecture is thus very straightforward and painless.

There are multiple ways how to include our cameras in your application.

The most natural one is to use the Simplon software package and its Simplon API library, which provides all the means for comfort work with the hardware and the acquired images. The sample code generator accompanying the library helps to integrate Simplon in your application with very limited effort.

Because the low level hardware access library is a full featured GenICam GenTL Producer, the entire functionality of all our camera models is readily available in any image processing software package supporting GenTL standard, such as Halcon, ActivVisionTools or Common Vision Blox.

Furthermore, the GigE Vision based cameras (PicSight-GigE and PicSight-Smart) are fully compatible also with those packages, that did not yet implement GenTL, including LabView, VisionPro or MIL (Matrox Imaging Library).

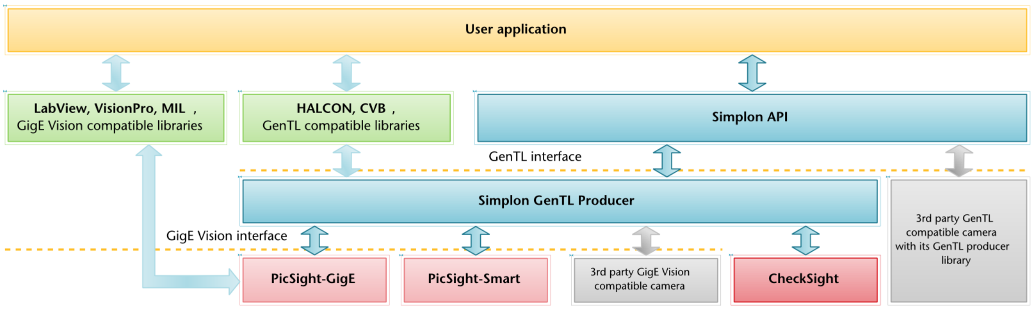

User application
LabView, VisionPro, MIL ,
GigE Vision compatible libraries
HALCON, CVB ,
GenTL compatible libraries
Simplon API
GenTL interface
Simplon GenTL Producer
3rd party GenTL compatible camera with its GenTL producer library
GigE Vision interface
PicSight-GigE
PicSight-Smart
3rd party GigE Vision compatible camera
CheckSight

Figure 2.1. Simplon connectivity capabilities

Note that thanks to Simplon's wide compatibility with GenTL and GigE Vision standards, we can also accommodate customers who need to combine our cameras with a 3rd party model in a single project — all in comfort of a single API.

2.2. Simplon GenTL Producer

The Simplon's hardware access layer unifies access to the entire camera product range, covering all interface and sensor types and exposes that through a single interface. The interface is compatible with the GenICam GenTL standard (being a full featured GenTL Producer) and can thus be connected to Simplon upper layers (Simplon API) or to any GenTL Consumer library.

Figure 2.2. Camera interfaces and sensors

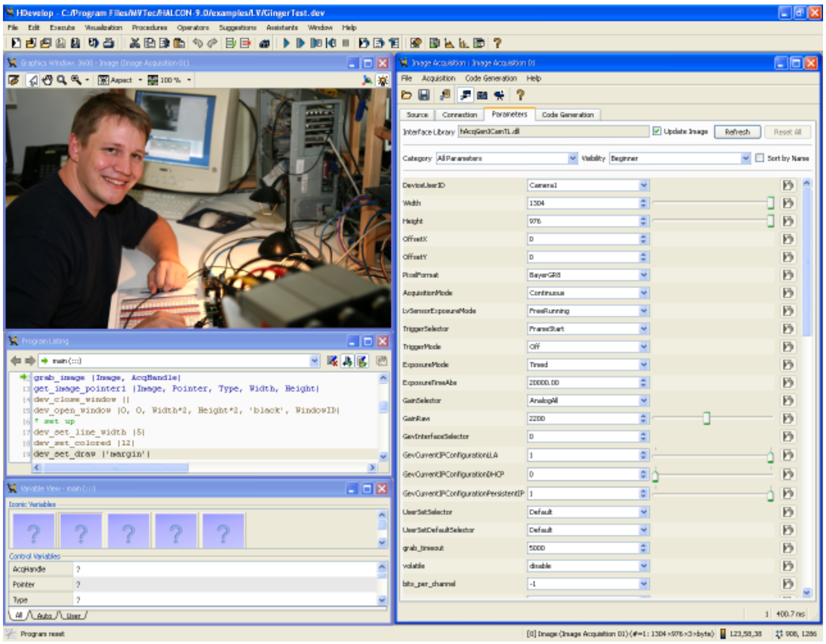
Simplon is one of the first few products claiming the recently established GenICam Transport Layer logo.

GEN<i>CAM
TRANSPORT LAYER

Simplon is also an officially certified as fully GigE Vision compatible software.

GiGE™
VISION

Following figure shows the Simplon GenTL Producer connected to the Halcon library.



HDevelop - C:/Program Files/MVTec/HALCON-9.0/examples/LV/GingerTest.dev
File Edit Execute Visualization Procedures Operators Suggestions Assistants Window Help
Graphics Window: 3600 - Image (Image Acquisition 01)
Aspect
100 %
Program Listing
main (:::)
Variable View - main (:::)
Iconic Variables
Control Variables
All Auto User
Program reset
Image Acquisition : Image Acquisition 01
File Acquisition Code Generation Help
Source Connection Parameters Code Generation
Refresh
Reset All

```
grab_image (Image, AcqHandle)
get_image_pointer1 (Image, Pointer, Type, Width, Height)
dev_close_window ()
dev_open_window (0, 0, Width*2, Height*2, 'black', WindowID)
* set up
dev_set_line_width (5)
dev_set_colored (12)
dev_set_draw ('margin')
```

AcqHandle	?
Pointer	?
Type	?

Interface Library: hAcqGenICamTL.dll ☑ Update Image

Category: All Parameters Visibility: Beginner □ Sort by Name

DeviceUserID	Camera1
Width	1304
Height	976
OffsetX	0
OffsetY	0
PixelFormat	BayerGR8
AcquisitionMode	Continuous
LvSensorExposureMode	FreeRunning
TriggerSelector	FrameStart
TriggerMode	Off
ExposureMode	Timed
ExposureTimeAbs	20000.00
GainSelector	AnalogAll
GainRaw	2200
GevInterfaceSelector	0
GevCurrentIPConfigurationLLA	1
GevCurrentIPConfigurationDHCP	0
GevCurrentIPConfigurationPersistentIP	1
UserSetSelector	Default
UserSetDefaultSelector	Default
grab_timeout	5000
volatile	disable
bits_per_channel	-1

1 400.7 ms

[0] Image (Image Acquisition 01) (#=1: 1304×976×3×byte) 123,58,38 908, 1286

Figure 2.3. Simplon GenTL Producer with Halcon

2.3. Simplon API

The diagram shows the basic architecture of the Simplon API library, showing the most commonly used components.

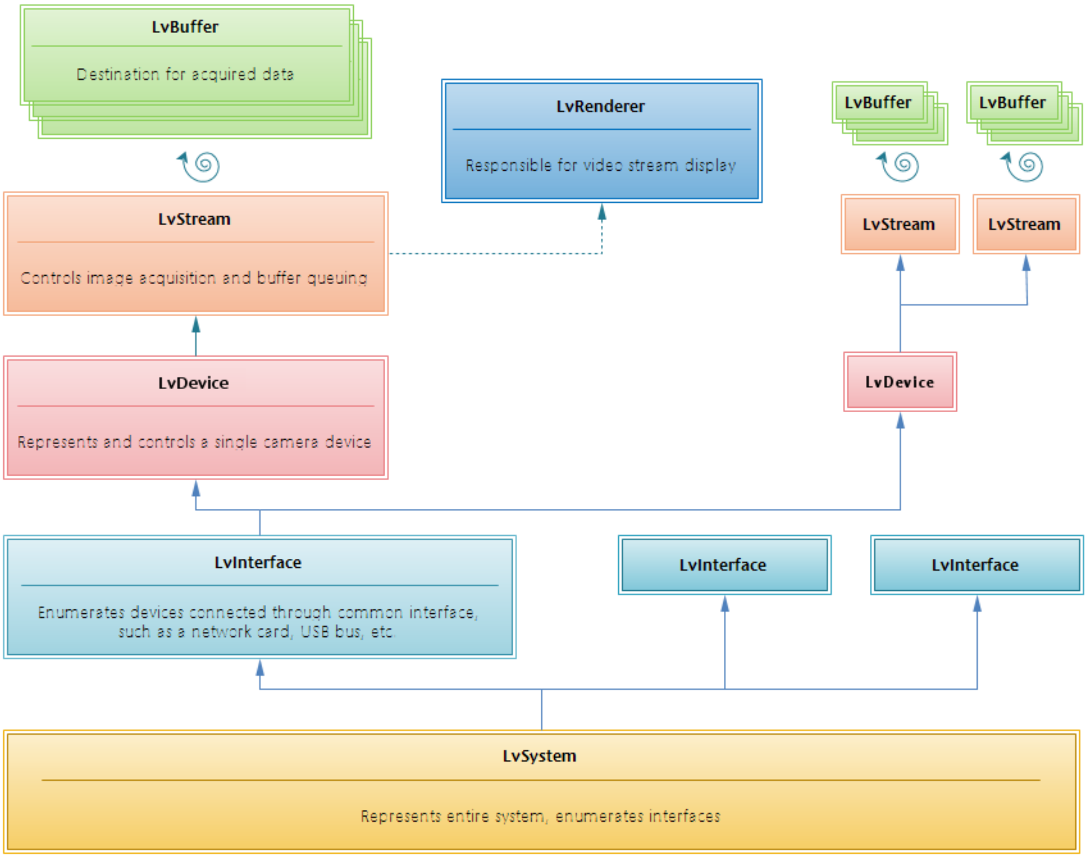

LvBuffer
Destination for acquired data
LvRenderer
Responsible for video stream display
LvBuffer
LvBuffer
LvStream
Controls image acquisition and buffer queuing
LvStream
LvStream
LvDevice
Represents and controls a single camera device
LvDevice
LvInterface
Enumerates devices connected through common interface, such as a network card, USB bus, etc
LvInterface
LvInterface
LvSystem
Represents entire system, enumerates interfaces

Figure 2.4. Main abstractions of the Simplon API and their relationships

The Simplon API functionality is available either through a pure C interface, through an object-oriented C++ interface or as a .Net class library.

It supports various use cases, from trivial applications, where most functionality, including configuration, threading or display is handled internally for you by the library to complex scenarios, where user has fine grain control over everything.

The main features of the Simplon API library were already outlined before.

Although the library provides the same functionality as any generic GenICam based package, Simplon API goes way further, providing direct access to all common camera features, shortcuts for easy handling of complex feature sets. Of course it handles all obvious acquisition related tasks such as buffer queuing, saving images etc.

Simplon API comes also with a preprocessing library that unifies access to preprocessing capabilities of some cameras and similar functionality in the software. When replacing a simple camera with a model capable of real-time preprocessing, the same user code will keep working seamless, only the task will get automatically offloaded from the host processor to the camera.

Last but not least, it also offers GUI components and dialogs for inclusion in user applications – such as the GenICam feature tree control.

While providing add-on features utilizing work with Leutron Vision cameras, the Simplon core is kept strictly generic, so that it works well with any GenICam compatible product.

2.4. Simplon Explorer and Simplon Source Code Generator

The Simplon Explorer tool is a primary testing and demonstration program built on top of the Simplon API — it provides easy access to the Simplon API and camera functionality, available right after the installation without any programming. Besides that, the Simplon Explorer offers additional built-in features designed to assist the developer to create a Simplon API application quickly and easily.

The user interface of the Simplon Explorer is discussed in Chapter 3, *Simplon Explorer in detail* [p. 22].

Writing applications utilizing all advantages of modern cameras, in particular if the application should stay generic, error resistant and open for cameras of different models (possibly even different technologies or different vendors) requires good understanding of principles behind the GenICam standard. And that usually means quite a steep learning curve, even if the details are wrapped in a well designed API.

To simplify and streamline the development process, Leutron Vision provides the Simplon Source Code Generator tool.

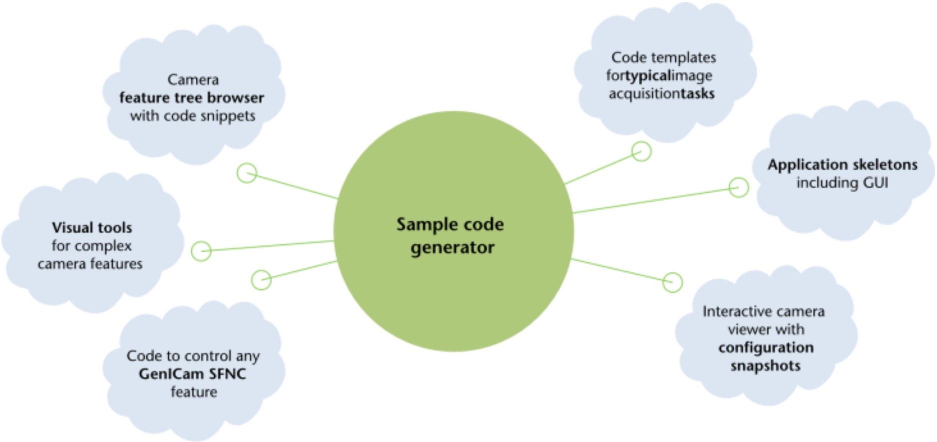
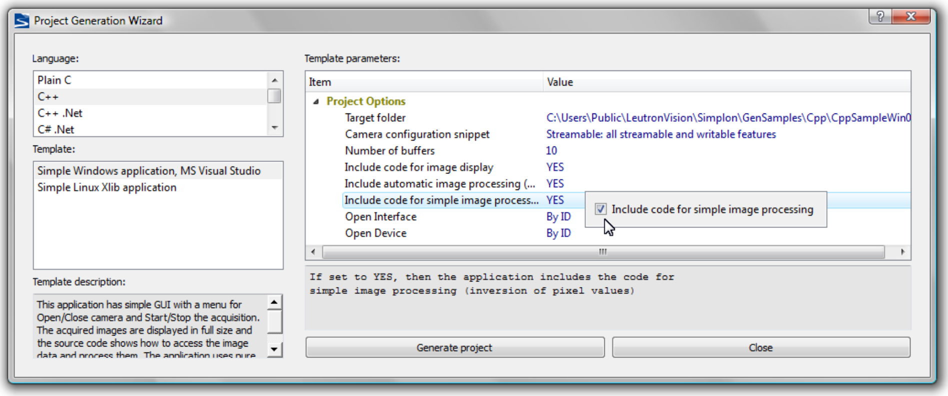


Figure 2.5. Simplon Source Code Generator features

- Single-line snippets to full-fledged examples including GUI.
- Templates for typical image acquisition and processing tasks, eg. camera enumeration, buffering techniques, HW/SW preprocessing etc.
- Browsing the camera's feature tree, generating code to control given feature. Browsing the feature tree including all standard SFNC features.
- Visual tools for complex camera features (trigger modes, complex digital I/O, LUT's and color transformations, AOI, etc.).
- Interactive camera configurator, snapshot & reuse full camera settings.
- C, C++, C++.Net, C# or VB.Net code. Operating system independent code. Optional GUI elements based on Qt or Win32 API (eventually Xlib).
- Interactive code generation, cut & paste the code to your application.
- Fully generic, works with any GenICam compatible camera.
- Generating customized code examples through several mouse clicks. The tool runs under Windows and Linux.



This application has simple GUI with a menu for Open/Close camera and Start/Stop the acquisition. The acquired images are displayed in full size and the source code shows how to access the image data and process them. The application uses pure

Item	Value
Project Options	
Target folder	C:\Users\Public\LeutronVision\Simplon\GenSamples\Cpp\CppSampleWin0
Camera configuration snippet	Streamable: all streamable and writable features
Number of buffers	10
Include code for image display	YES
Include automatic image processing (...	YES
Include code for simple image process...	YES
Open Interface	By ID
Open Device	By ID

☑ Include code for simple image processing

```
If set to YES, then the application includes the code for
simple image processing (inversion of pixel values)
```

Figure 2.6. Simplon Source Code Generator configuration

Details about the Simplon Source Code Generator with guidelines how to use it to quickly develop an application with Simplon API are provided in Simplon programmer's guide.

2.5. Crucial industry standards and Leutron Vision's role in their development

Leutron Vision plays a major role in developing, supporting and propagating the principal standards in the machine vision industry today — GenICam and GigE Vision.

Our goal is to keep supplying products fully compatible with all the established standards and work to maintain the standards to be prepared for new demands of tomorrow's applications.

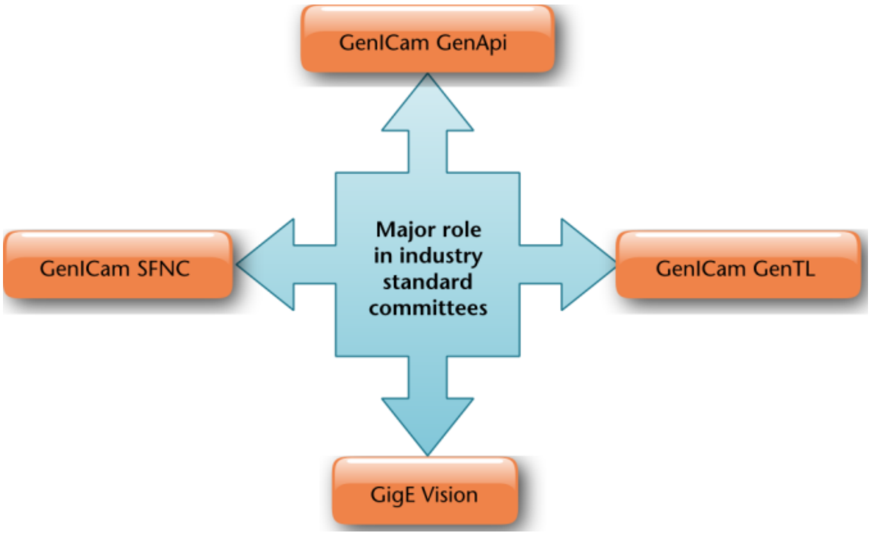


Figure 2.7. Leutron Vision activity in standardization committees

Leutron Vision comes with the innovative approach to use the GenICam GenTL interface as a middle layer of the camera SDK. This allows maximum flexibility and both-side openness of the library. The customer gains great deal of independence and scalability.

GenICam GenApi — GenICam API, defining means to describe all camera features, their corresponding control registers and providing access to those features

GenICam GenTL — GenICam GenTL (Transport Layer), an interface between applications and camera-control libraries, covering device enumeration, image acquisition and related tasks.

Copyright © Leutron Vision

GenICam SFNC — SFNC (Standard Features Naming Convention), standardizing names (ID's), types and other aspects for most common features of a machine vision camera

GigE Vision — Set of UDP/IP based protocols standardizing access to Gigabit Ethernet based cameras and devices.

Deeper information about these industrial standards is provided in Chapter 8, *Vision Standards* [p. 67].

2.6. Documentation & customer support

We are investing big effort to keep all product documentation up to date. The entire documentation set is built automatically every night, including the reference documentation and technical specifications extracted from the master product database.

Our support team is willing to answer all your questions and resolve potential issues or special requests in shortest possible time.

Figure 2.8. Leutron Vision support

2.7. Licensing information

The Simplon package and/or the software included in the cameras is using following 3rd party software components:

- GenICam GenApi reference implementation — distributed under the GenICam license by the GenICam committee. The copyright is held by the GenICam committee and its contributing members working group (which includes Leutron Vision). The package as well as the license is available from www.genicam.org[1]. The GenICam GenApi reference implementation in turn uses Apache Xalan-C++ and Xerces-C++ libraries (Apache license, www.apache.org[2]) and the modified MathParser librariy (LGPL license, kirya.narod.ru/mathparser.html[3]).
- zlib compression library: available from www.zlib.net[4] under zlib license.

[1] http://www.genicam.org
[2] http://www.apache.org/
[3] http://kirya.narod.ru/mathparser.html
[4] http://www.zlib.net/

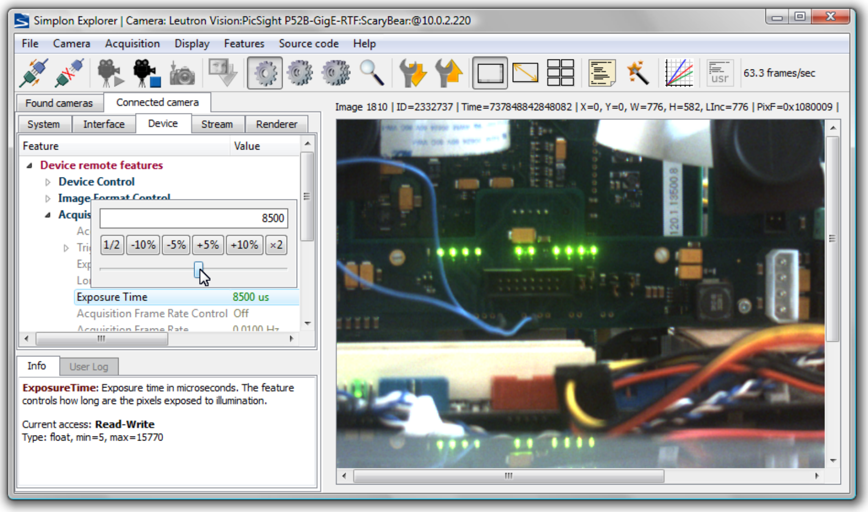
- lwIP (lightweight TCP/IP stack): available from savannah.nongnu.org/projects/lwip[5] under modified BSD license. The copyright © is held by Swedish Institute of Computer Science.
- The Independent JPEG Group's JPEG software (libjpeg): available from www.ijg.org[6] (Independent JPEG Group).
- OpenCV: available from opencv.willowgarage.com[7] under BSD license.

[5] http://savannah.nongnu.org/projects/lwip/
[6] http://www.ijg.org/
[7] http://opencv.willowgarage.com/

3. Simplon Explorer in detail

The Simplon Explorer is a demonstration and developer-aid tool delivered with Simplon. It is written fully using the Simplon API and therefore can serve as demo for both the camera and Simplon API capabilities.

In Windows, the Simplon Explorer can be started from the system menu: Start → Programs → Leutron Vision Simplon → Simplon Explorer.



Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220

63.3 frames/sec

Image 1810 | ID=2332737 | Time=737848842848082 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0x1080009 |

Feature	Value
Device remote features	
Device Control	
Image Format Control	
Acquis	
Exposure Time	8500 us
Acquisition Frame Rate Control	Off
Acquisition Frame Rate	0.0100 Hz

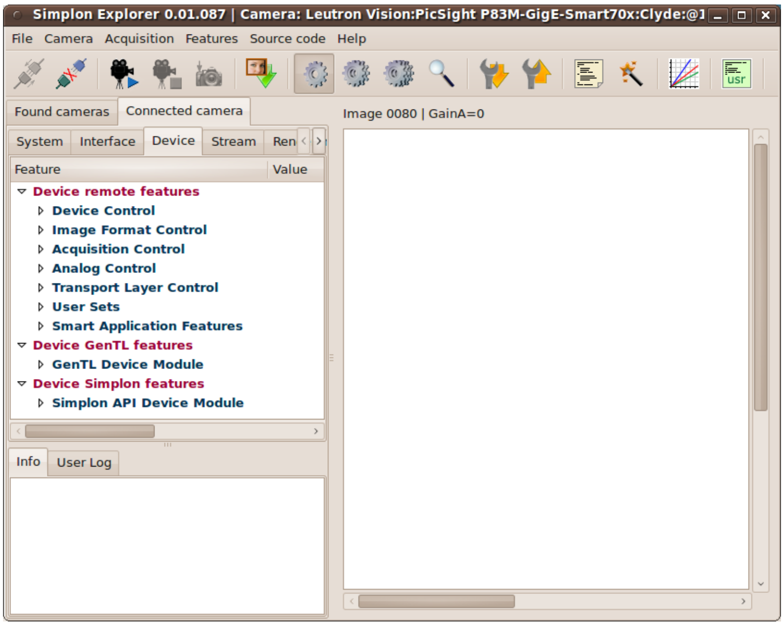
8500

ExposureTime: Exposure time in microseconds. The feature controls how long are the pixels exposed to illumination.

Current access: **Read-Write**
Type: float, min=5, max=15770

Figure 3.1. Simplon Explorer running under Windows

In Linux, it can be started for example by executing the `lv.explorer` command from the shell, the installer attempts to add Simplon executables to the system path. If not, it can still be invoked using full path: `/opt/simplon/bin/lv.explorer`.



Simplon Explorer 0.01.087 | Camera: Leutron Vision:PicSight P83M-GigE-Smart70x:Clyde:@1

Image 0080 | GainA=0

Feature	Value
Device remote features	
Device Control	
Image Format Control	
Acquisition Control	
Analog Control	
Transport Layer Control	
User Sets	
Smart Application Features	
Device GenTL features	
GenTL Device Module	
Device Simplon features	
Simplon API Device Module	

Figure 3.2. Simplon Explorer running under Linux

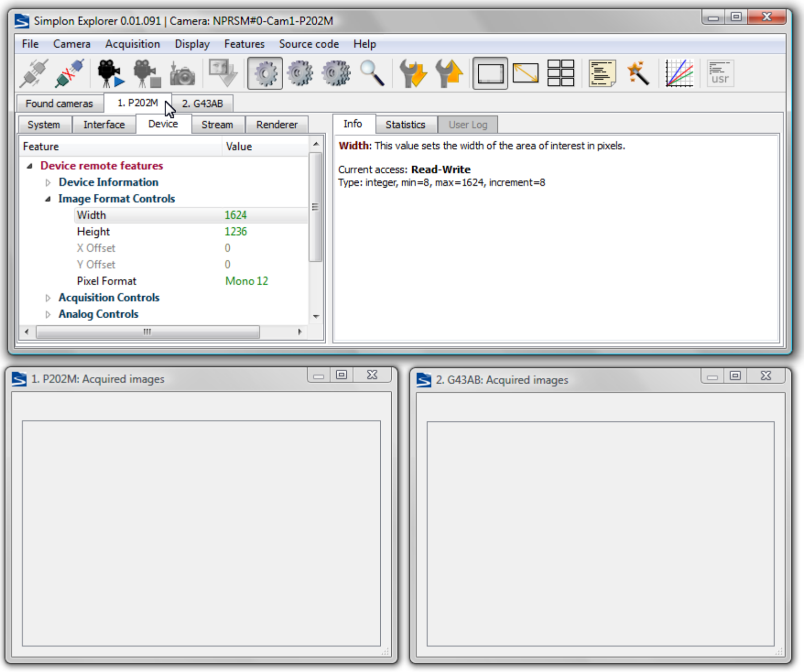
In the next text we will show only screen shots from Windows; in Linux they would be identical (with the difference only in the display options).

Simplon Explorer offers its functionality in a set of menus. Many menu items are duplicated as toolbar buttons, for convenience. In the screen shots we will show mostly the usage of the toolbar buttons.

Simplon Explorer can be run in:

- **Single camera (compact) mode**, which uses only single application window. In this mode only one camera can be opened.
- **Multiple cameras mode**, which uses one window for controls and one window for each connected camera. In this mode you can connect multiple cameras and use them concurrently.

The mode can be set in the Settings dialog (see Section 3.5, "Settings" [p. 31]) and is applied when the Explorer is started next time.



Feature	Value
Device remote features	
Device Information	
Image Format Controls	
Width	1624
Height	1236
X Offset	0
Y Offset	0
Pixel Format	Mono 12
Acquisition Controls	
Analog Controls	

Width: This value sets the width of the area of interest in pixels.

Current access: **Read-Write**
Type: integer, min=8, max=1624, increment=8

Figure 3.3. Simplon Explorer running in the Multiple cameras mode

3.1. Connecting and disconnecting the camera

First step after starting the tool is to select and connect a camera. Simplon Explorer uses regular Simplon approach to enumerate cameras — depending on the configuration it can offer cameras from a single or from all GenTL Producer libraries available on the system. By default, it connects to the Simplon GenTL Producer, which handles all Leutron Vision cameras (PicSight and CheckSight families) as well as 3rd party GigE Vision compatible cameras.

All the found cameras are displayed in a tree on the *Found Cameras* tab. The tree lists all found *Systems* (GenTL producers), under each System it lists all *Interfaces*, containing at least one active device (camera), and under each Interface it lists all found *Devices*. When you select a device, you can connect it by clicking on the *Connect camera* button, or simply by double-clicking on the Device.

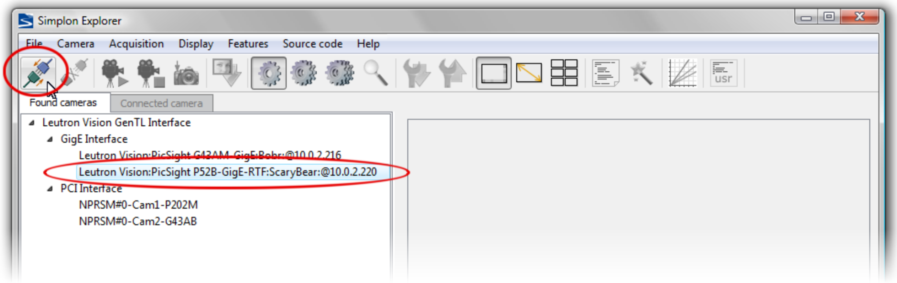

Simplon Explorer
File Camera Acquisition Display Features Source code Help
Found cameras
Connected camera
Leutron Vision GenTL Interface
GigE Interface
Leutron Vision:PicSight G43AM-GigE:Bobr:@10.0.2.216
Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
PCI Interface
NPRSM#0-Cam1-P202M
NPRSM#0-Cam2-G43AB

Figure 3.4. Simplon Explorer: Connecting a camera

If the desired camera is not available, check again, if it is properly connected and powered. For GigE Vision cameras, check also the network configuration and firewall settings. Simplon Explorer checks for the cameras existence at startup. You can also update the camera list during the Explorer run — use the *Update camera list(s)* menu item.

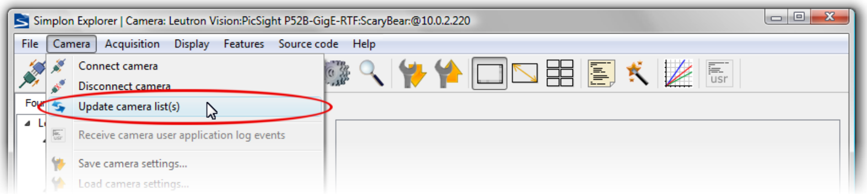

Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
File Camera Acquisition Display Features Source code Help
Connect camera
Disconnect camera
Update camera list(s)
Receive camera user application log events
Save camera settings...
Load camera settings...

Figure 3.5. Simplon Explorer: Updating the camera list

Simplon Explorer connect the camera in *exclusive access* mode, that means other applications cannot connect to this camera, while is connected by the Simplon Explorer. You can disconnect the camera and release it for other applications using the *Disconnect camera* button. Remember to do so namely when you come to the *Source code wizard*, where the generated application will need to connect to the same camera.

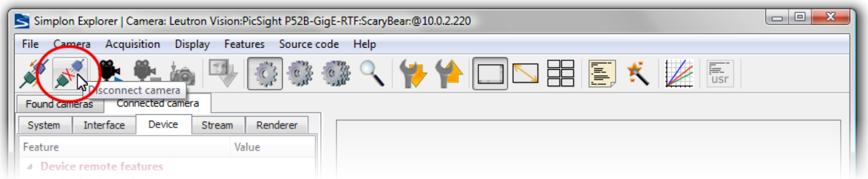
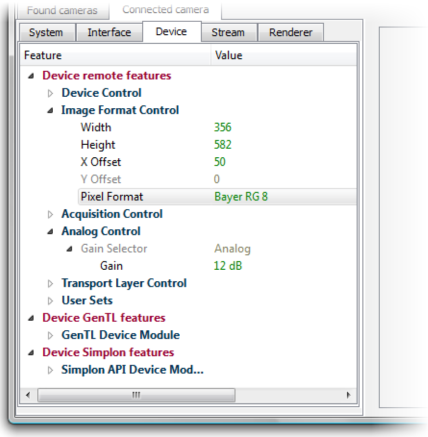

Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
File Camera Acquisition Display Features Source code Help
Disconnect camera
Found cameras
Connected camera
System Interface Device Stream Renderer
Feature
Value
Device remote features

Figure 3.6. Simplon Explorer: Disconnecting the camera

3.2. Features

Having the camera open, its **feature tree** is displayed, allowing to configure the camera. Actually, you will see 5 feature trees: from *System*, *Interface*, *Device*, *Stream* and *Renderer*. The role of these trees is explained in detail in the Simplon programmer's guide, for the first steps use only the *Device* tab with the tree, containing the most important features.



Feature	Value
◢ Device remote features	
▷ Device Control	
◢ Image Format Control	
Width	356
Height	582
X Offset	50
Y Offset	0
Pixel Format	Bayer RG 8
▷ Acquisition Control	
◢ Analog Control	
◢ Gain Selector	Analog
Gain	12 dB
▷ Transport Layer Control	
▷ User Sets	
◢ Device GenTL features	
▷ GenTL Device Module	
◢ Device Simplon features	
▷ Simplon API Device Mod...	

Figure 3.7. Simplon Explorer: Device feature tree

When you browse the features in the feature tree, you can see that some of them are greyed out — these are the *read-only* features. Other features can be modified — these are *read-write* features. This status need not be permanent, for example some features become read-only during the acquisition running (like the *Width* and *Height* device features).

You can also notice that features are of several types:

- a **Boolean** value — represented by a check box (for example *Chunk Mode Active*)
- an **integer** number — represented by a compound control with edit box, slider, inc/dec buttons (for example image *Width* and *Height*)
- a **float** number — also represented by a compound control with edit box, slider, inc/dec buttons (for example *Exposure Time*)
- an **enumeration** — represented by a combo box with available choices (for example *Pixel Format*)
- a **string** — represented by an edit box (for example *Device User ID*)
- a **command** — represented by a button for command execution (for example *User Set Load*)

3.2.1. Beginner-Expert-Guru level

Features are classified to 3 levels: *Beginner*, *Expert* and *Guru*. While the Beginner level (displayed as default) contains only a small subset of the most commonly used (and well understandable) features, the Guru level on the other hand includes all available features. In the menu or on the toolbar, you can switch to the Expert or Guru level, where a larger subset or all features are displayed.

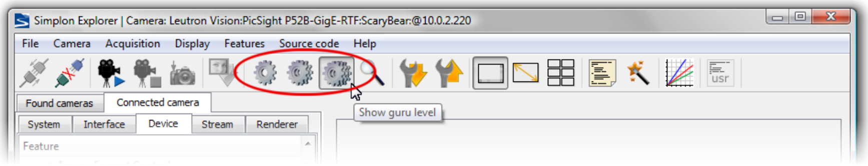
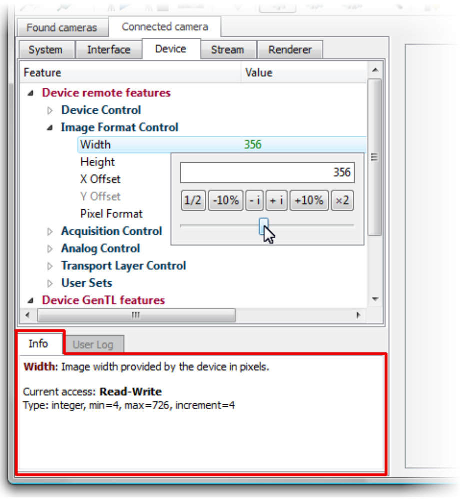


Figure 3.8. Simplon Explorer: Feature levels

As you can see, in the Guru level the device provides quite a high number of features. We recommend to use the *Beginner* level features only for initial experiments and switch to the *Expert* level, when doing more advanced tests. The *Guru* level should be avoided, since features in this level should not be usually touched during regular operation and altering them can lead to unexpected results.

3.2.2. Help for a feature

In a set of hundreds of features it may not be easy to guess the purpose of each feature. For this reason, the device provides also a **description** of each feature. In the *Info panel* Explorer shows the description for each selected feature:



Width: Image width provided by the device in pixels.

Current access: **Read-Write**
Type: integer, min=4, max=726, increment=4

Figure 3.9. Simplon Explorer — The info panel

3.2.3. Configuring the device

Typically, you first configure the device parameters and then start the acquisition. In the tree, you can configure any read-write feature, simply by clicking on it — a control window pops up, enabling you to change the feature. Note that the availability of one feature may depend on another — for example when the Image *Width* is set to *Max Width*, then the *X Offset* is not writable, because the *Width* + *XOffset* must be less or equal to *Max Width*, so there is no space to increase the *X Offset*. However, when you make the *Width* smaller, you will see that the *X Offset* becomes writable and that its maximum value follows the *Max Width* – *Width* formula. Furthermore, the availability of the offsets can be dependent on current *AOI Mode*. So in some cases the dependencies may not be understandable at the first sight, but you can usually discover them fast by playing around with the features.

The tree of camera features follows the principles coming from the GenICam standard. Its subtleties are discussed in a dedicated chapter, Section 4.1, “Understanding the feature tree” [p. 34]. Getting familiar with that chapter will help you in effective work with the camera.

Furthermore, the device configuration is also described in detail in the PicSight GigE Manual and CheckSight Manual.

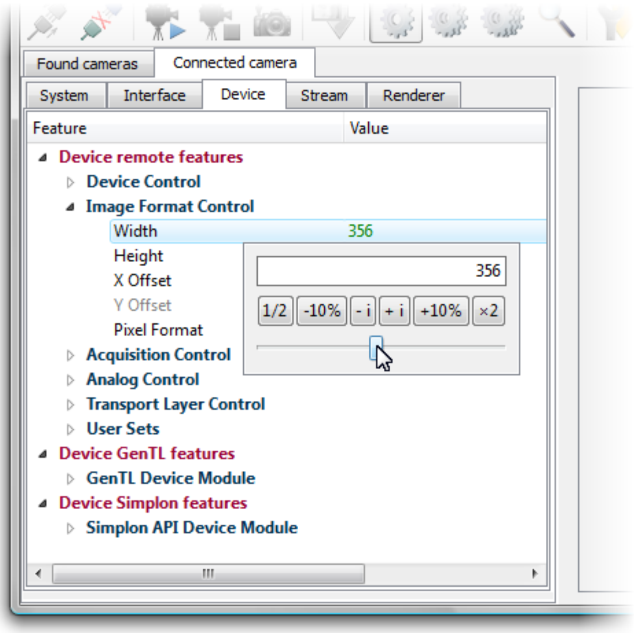


Figure 3.10. Simplon Explorer — Setting a feature (Image Width)

Also note, that some features have so called *selectors*. For example the device can have several I/O ports and to configure a specific port, you should select it by the *Line Selector* and then the features in the tree under this selector are used for the selected port.

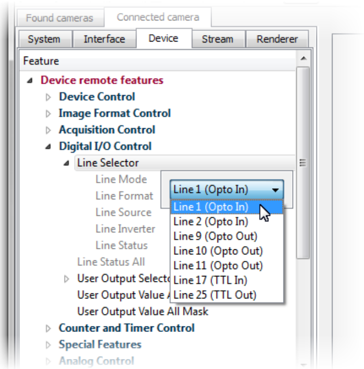


Figure 3.11. Simplon Explorer — Using a selector (Line)

3.2.4. Finding a feature

In a complex feature tree it might be difficult to find desired feature. Simplon Explorer offers a tool for finding a feature if you enter a part of its *name* or *display name*.

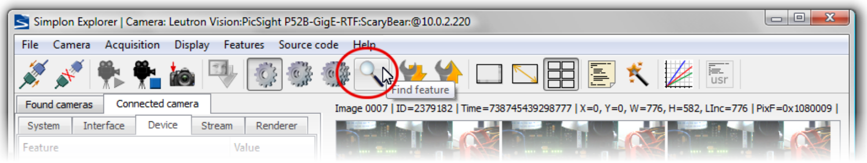
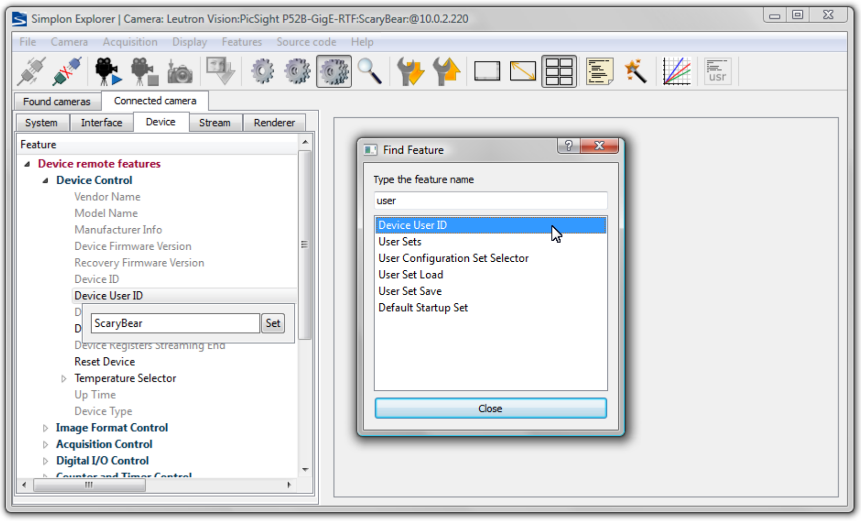

Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220

Image 0007 | ID=2379182 | Time=738745439298777 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0x1080009 |

Figure 3.12. Simplon Explorer — Find feature

During typing the feature name Simplon Explorer updates a list of all features, in which the written substring occurs. By clicking on a feature in the list, the feature is focused in the tree.


Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220

- Device remote features
  - Device Control
    - Vendor Name
    - Model Name
    - Manufacturer Info
    - Device Firmware Version
    - Recovery Firmware Version
    - Device ID
    - Device User ID: ScaryBear [Set]
    - Device Registers Streaming End
    - Reset Device
    - Temperature Selector
    - Up Time
    - Device Type
  - Image Format Control
  - Acquisition Control
  - Digital I/O Control
  - Counter and Timer Control

Find Feature

Type the feature name: user

- Device User ID
- User Sets
- User Configuration Set Selector
- User Set Load
- User Set Save
- Default Startup Set

Figure 3.13. Simplon Explorer — Find feature dialog

3.2.5. Saving and restoring the configuration

It is possible to store full camera configuration (= all needed features) at any time to a file. That configuration can be then later reloaded either again in the Simplon Explorer or in the user application (the Simplon API provides a function to load settings). Using this functionality, the programmer does not need to care about the possible feature dependencies, Simplon API handles that correctly. Furthermore, the stored configuration can be used for multiple connected cameras, provided that they are of the same type and firmware version.

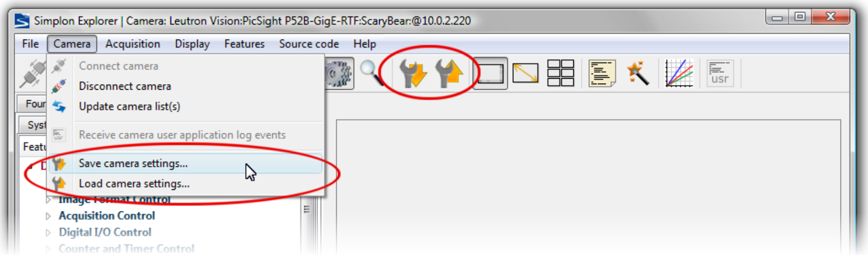
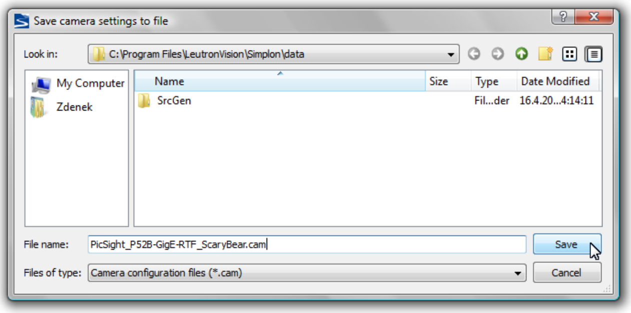

Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220

Figure 3.14. Simplon Explorer: Save camera settings


Save camera settings to file

Look in: C:\Program Files\LeutronVision\Simplon\data

Name	Size	Type	Date Modified
SrcGen		Fil...der	16.4.20...4:14:11

File name: PicSight_P52B-GigE-RTF_ScaryBear.cam

Files of type: Camera configuration files (*.cam)

Figure 3.15. Simplon Explorer: Save camera settings dialog

3.3. Start and stop the acquisition

Having the camera configured, you can start the acquisition using the *Start acquisition* button . If the acquisition does not start as expected, review again the camera feature settings. For GigE Vision cameras, once again, the network configuration and firewall settings are to be rechecked.

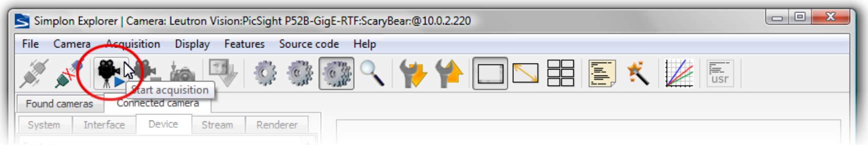


Figure 3.16. Simplon Explorer: Start acquisition

In case you have configured the camera for triggered acquisition and the *Software Trigger* feature becomes available, you can trigger the camera by the button in the toolbar:

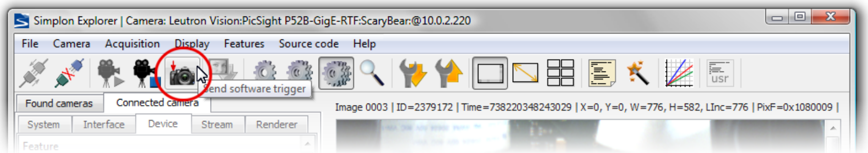


Image 0003 | ID=2379172 | Time=738220348243029 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0x1080009 |

Figure 3.17. Simplon Explorer: Trigger the camera

Note that some features become read-only when you start the acquisition, for example you cannot change the image size during the acquisition. And vice-versa: other features may become usable only after you start the acquisition — for example the *Trigger Software* command, if you use the triggered mode.

The acquisition can be stopped through the *Stop acquisition* button.

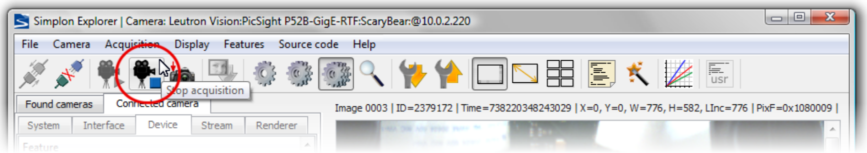


Image 0003 | ID=2379172 | Time=738220348243029 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0x1080009 |

Figure 3.18. Simplon Explorer: Stop acquisition

3.4. Display of images

The way how Simplon Explorer displays the acquired images can be configured — the available options are **full size** (if the image does not fit in the display area, scroll bars are added), **scale to fit** (image is scaled to fit to the display area) and **tiled display** (Simplon Explorer displays series of consecutive images in tiles). Note that the *scale to fit* and *tile* modes might not be available in Linux.

To be configured through the Display menu or corresponding buttons.

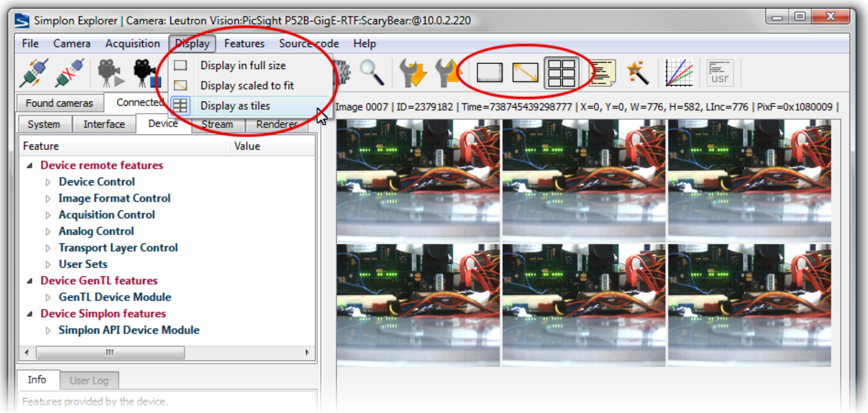

Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
File Camera Acquisition Display Features Source code Help
Display in full size
Display scaled to fit
Display as tiles
Found cameras Connected
System Interface Device Stream Renderer
Feature Value
Device remote features
Device Control
Image Format Control
Acquisition Control
Analog Control
Transport Layer Control
User Sets
Device GenTL features
GenTL Device Module
Device Simplon features
Simplon API Device Module
Info User Log
Features provided by the device.
Image 0007 | ID=2379182 | Time=738745439298777 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0x1080009 |

Figure 3.19. Simplon Explorer: Display modes

3.4.1. Automatic image preprocessing

Simplon Explorer makes automatic image preprocessing. This includes namely the **Bayer decoding** for color cameras, applying the **LUT** (Lookup Table), to which can be added parameters like *white balance*, *gamma*, *brightness* and *contrast*, and the **color correction**, by which can be adjusted the *saturation*. Simplon automatically determines, if the required functionality is available in camera hardware, and if not, does the necessary operations by software (this increases the CPU load).

The automatic processing is by default switched on. You can switch it off in menu:

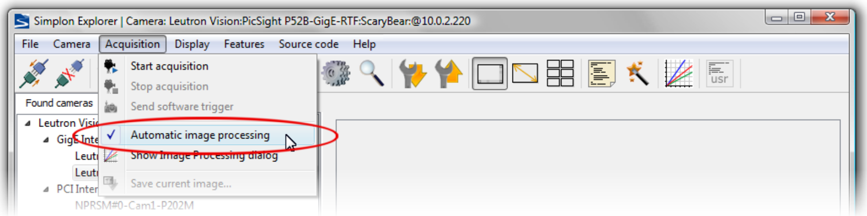

Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
File Camera Acquisition Display Features Source code Help
Start acquisition
Stop acquisition
Send software trigger
✓ Automatic image processing
Show Image Processing dialog
Save current image...
Found cameras
Leutron Visi
GigE Inte
Leuti
Leuti
PCI Inter
NPRSM#0-Cam1-P202M

Figure 3.20. Simplon Explorer: Automatic image processing

You can also display a dialog, where you can adjust the processing parameters:

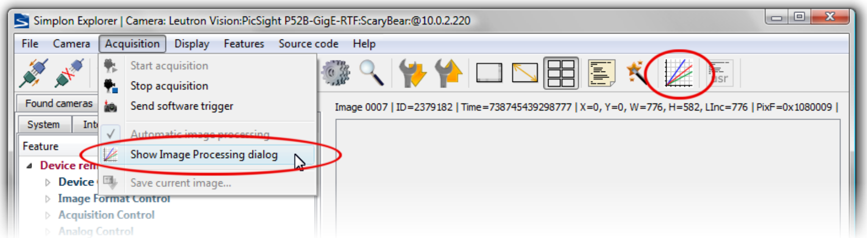
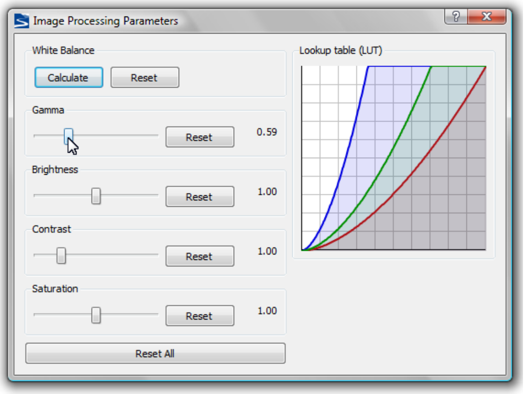

Simplon Explorer | Camera: Leutron Vision:PicSight P52B-GigE-RTF:ScaryBear:@10.0.2.220
File Camera Acquisition Display Features Source code Help
Start acquisition
Stop acquisition
Send software trigger
Automatic image processing
Show Image Processing dialog
Save current image...
Found cameras
System Int
Feature
Device rem
Device
Image Format Control
Acquisition Control
Analog Control
Image 0007 | ID=2379182 | Time=738745439298777 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0x1080009 |

Figure 3.21. Simplon Explorer: Show image processing dialog



Gamma 0.59

Brightness 1.00

Contrast 1.00

Saturation 1.00

Figure 3.22. Simplon Explorer: Image processing dialog

Note that the *white balance* factors are always calculated from the *next* acquired image, so after pressing *Calculate* button you will not see a change until a new image is acquired.

3.4.2. Saving acquired image

When you stop the acquisition, you can save the last acquired image, to a BMP, TIFF or JPEG file. If you have the automatic processing switched on, the processed image is saved, otherwise the original image is saved. Before saving, the image is converted to suitable pixel format, for example when you save an image with a 12–bit mono pixel format to a BMP or JPEG file, it is automatically converted to 8–bit mono pixel format; to a TIFF file it is converted to 16–bit mono pixel format.

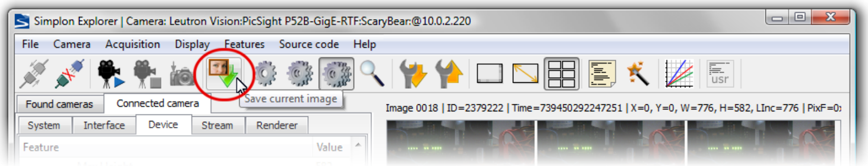
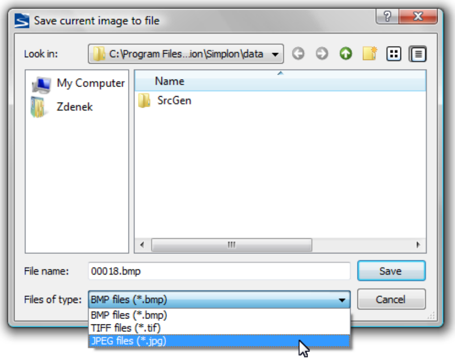


Image 0018 | ID=2379222 | Time=739450292247251 | X=0, Y=0, W=776, H=582, LInc=776 | PixF=0:

Figure 3.23. Simplon Explorer: Save image

Select the image format in the Save image dialog:



Look in: C:\Program Files...ion\Simplon\data

File name: 00018.bmp

Files of type: BMP files (*.bmp)

Figure 3.24. Simplon Explorer: Save image dialog

3.5. Settings

Simplon Explorer settings are available in the menu:

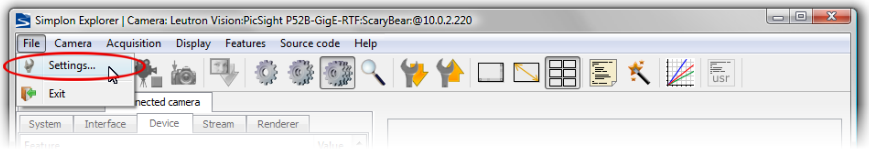
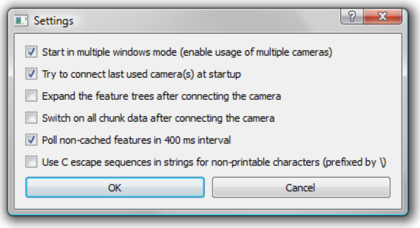


Figure 3.25. Simplon Explorer: Settings

The following settings are available:


Settings

☑ Start in multiple windows mode (enable usage of multiple cameras)

☑ Try to connect last used camera(s) at startup

□ Expand the feature trees after connecting the camera

□ Switch on all chunk data after connecting the camera

☑ Poll non-cached features in 400 ms interval

□ Use C escape sequences in strings for non-printable characters (prefixed by \)

Figure 3.26. Simplon Explorer: Settings dialog

- **Start in multiple windows mode (enable usage of multiple cameras)**. If ON, Simplon Explorer starts in multiple window mode, enabling concurrent usage of multiple cameras. If OFF, it starts in a compact, single window mode, enabling to connect one camera only. This settings applies to the next Simplon Explorer start (does not have an immediate effect).

- **Try to connect last used camera at startup**. If you close Simplon Explorer without explicitly disconnecting the camera and this option is ON, the Explorer will attempt to automatically connect the same camera next time it is run (if the camera is still available).

- **Expand the feature trees after connecting the camera**. By default the trees are displayed as *collapsed*, that means only the top level items are displayed. When you set this option to ON, all the trees are fully expanded when the camera is connected.

- **Switch on all chunk data after connecting the camera**. If the camera delivers *chunk data* with each image and this option is ON, the Explorer automatically switches on all available chunk data features. If you do not use this option, you can always switch on selected chunk data features manually. Some of the chunk data are displayed automatically in a line above the image.

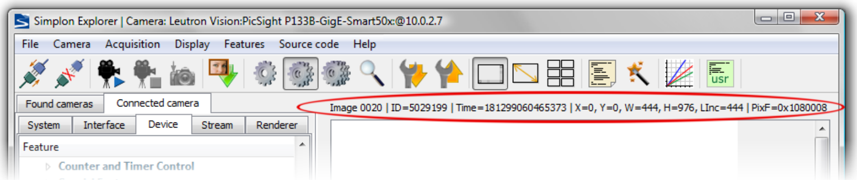


Image 0020 | ID=5029199 | Time=181299060465373 | X=0, Y=0, W=444, H=976, LInc=444 | PixF=0x1080008

Figure 3.27. Simplon Explorer: Chunk data

- **Poll non-cached features in 400 ms interval**. If set to ON, the Explorer polls the non-cached features (such as *Device Temperature* or *Up Time*) with the minimum period of 400 ms and updates their values in the feature tree.

- **Use C escape sequences in strings for non-printable characters**. In case some of the string features need to use a non-printable characters (for example strings sent via RS-232 port), you can switch this option ON to be able to write and see such characters. Then a C language escape sequence syntax is used for non-printable characters. The sequence begins with a

backslash followed by a **single letter** (r = Carriage Return, n = Line Feed, etc.), or a number in form of **x + 2 hexadecimal digits** or **3 decimal digits**. The number expresses the character ordinal number. A backslash itself must be written as double backslashes.

```
  \ + letter (b, t, n, v, f, r)
  \ + xNN hexadecimal number
  \ + NNN decimal number

  \\ = \x5C = \092   (backslash)
  \b = \x08 = \008   (backspace)
  \t = \x09 = \009   (tab)
  \n = \x0A = \010   (line feed)
  \v = \x0B = \011   (vertical tab)
  \f = \x0C = \012   (form feed)
  \r = \x0D = \013   (carriage return)
```

3.6. Source code generator

Simplon Explorer also provides a powerful set of tools for generating the source code. The detailed information about the developer assistive tools with guidelines how to use them during development is provided in the Simplon source code generator chapter in the Simplon programmer's guide.

4. Important principles behind Simplon interface

4.1. Understanding the feature tree

The “feature tree” is one of the most important patterns used within Simplon. It is a tree of inter-dependent features used to configure individual hardware and Simplon components, with a GenICam GenApi mechanism running under the hood. It is important to understand different aspects of the feature tree functionality to be able to configure the system effectively and reliably.

4.1.1. Feature tree instances

The feature tree paradigm is reused to configure the devices and various software components. It is important to understand that the set of supported features can vary significantly among different hardware (camera) models or even among different revisions of the same model.

Simplon Explorer displays all available feature tree instances, allowing to configure the entire system intuitively. The guide how to control individual features programmatically is provided in Simplon programmer's guide.

Remote device features

Set of features used to configure every single device connected to the system. This feature tree is designed by the device manufacturer (by means of the GenICam standard) and Simplon just exposes it to the user. The “device” is typically a camera, but it can be any configurable device connected to the system.

GenTL Producer features

Features configuring the GenTL Producer, which can be either directly Simplon GenTL Producer or a 3rd party GenTL Producer controlling cameras of another vendor. There is one feature tree available for each distinct component describing the acquisition system:

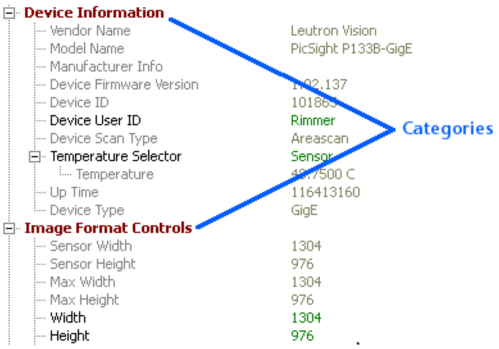
- **System.** Represents a GenTL Producer library. In most cases, there will be just a single system module available — representing the Simplon GenTL Producer.
- **Interface.** Describes a physical interface used to connect cameras, for example a network segment interfacing GigE Vision cameras or a bus interfacing the CheckSight camera modules.
- **Device.** Software representation of a device connected to the system.
- **Data stream.** Controls the stream of the image data coming from the device. In most cases there will be just a single data stream per device, but there might be devices featuring multiple data streams.
- **Buffer.** Allows to query information about individual image buffers.

Simplon API features

On top of the configuration options provided by the camera itself (and eventually the supporting GenTL Producer library), Simplon API itself also adds a rich set of features, which are, for the sake of consistency, also controlled through a feature tree. Again, Simplon would provide a separate feature tree per component.

4.1.2. Categories

The features are sorted into categories, grouping together related parameters. The only purpose of categories is to improve the visual representation of a complex feature tree. From programmer's perspective (access of individual features), the categories are not important.



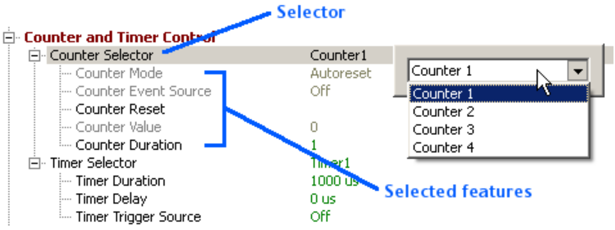
```
⊟ Device Information
    Vendor Name                 Leutron Vision
    Model Name                  PicSight P133B-GigE
    Manufacturer Info
    Device Firmware Version     1.02.137
    Device ID                   101865
    Device User ID              Rimmer
    Device Scan Type            Areascan
  ⊟ Temperature Selector        Sensor
      Temperature               48.7500 C
    Up Time                     116413160
    Device Type                 GigE
⊟ Image Format Controls
    Sensor Width                1304
    Sensor Height               976
    Max Width                   1304
    Max Height                  976
    Width                       1304
    Height                      976
```

4.1.3. Selectors

Selectors are features controlling sets or arrays of identical features.

A selector can be viewed as an analogy of an array index. Similarly as changing an array index does not by itself modify any value stored in the array, changing a selector does **never** modify the selected features — it **does not** alter the actual camera status.

For example, switching the `LineSelector` value does not at all change configuration of the I/O lines. It just selects the line to be configured through the "selected" features, such as `LineSource` or `LineInverter`.

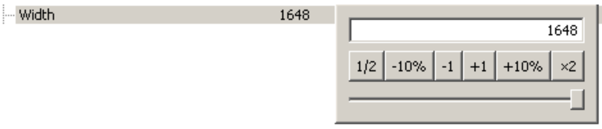


```
⊟ Counter and Timer Control
  ⊟ Counter Selector            Counter1     [Counter 1 ▼]
      Counter Mode              Autoreset     Counter 1
      Counter Event Source      Off           Counter 2
      Counter Reset                           Counter 3
      Counter Value             0             Counter 4
      Counter Duration          1
  ⊟ Timer Selector              Timer1
      Timer Duration            1000 us
      Timer Delay               0 us
      Timer Trigger Source      Off
```

4.1.4. Feature interface types

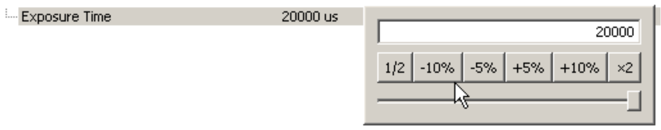
Depending on the purpose of respective features, they can be represented through different interfaces.

Integer
: Up to 64-bit integer value, which can have assigned min/max values and an increment (or a set of allowed values). The feature values must respect these constraints — if the application tries to set an invalid value, Simplon API readjusts it to the closest valid one.



```
Width                1648      [                1648 ]
                               [1/2][-10%][-1][+1][+10%][x2]
```

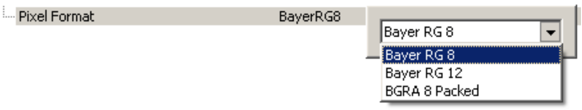
Float
: Floating point number with optional min/max (or set of allowed values). For visualization purposes (slider in a GUI representation), the floating point features can also have an increment, but unlike with integers, setting the values aligned with the increment is not forced. The camera can, however, realign the value internally to a closest valid discrete value — the application can re-read the feature after setting to know the actually used exact value.


Exposure Time 20000 us

20000

Enumeration

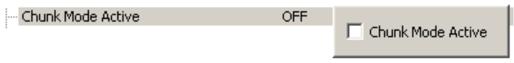
Feature allowing to select from given set of entries. Some of the entries might have self-clearing behaviour — they reset automatically to another entry. This behaviour is similar to a command feature.


Pixel Format BayerRG8

Bayer RG 8

Boolean

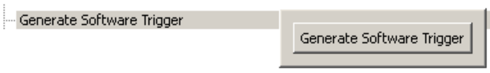
Simple flags that can have only two values, `true` or `false`.


Chunk Mode Active OFF

□ Chunk Mode Active

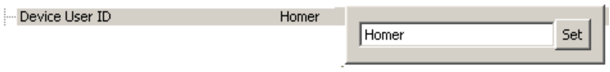
Command

Purpose of the command features is to init various actions (start acquisition, calibration, automatic white balance etc.). The command feature provides feedback when the action is finished.


Generate Software Trigger

String

String (character) based features.


Device User ID Homer

Homer

Register

Memory blob with unspecified representation. Can be used for example to load/save the entire LUT in a single feature access.

4.1.5. Feature properties

The features have also additional properties, some of them dynamic, which can change during operation, others static (remain fixed during runtime). Simplon API provides access to all these properties.

4.1.5.1. Dynamic properties

Access mode

The access mode specifies, whether the feature is available for reading and/or writing. It is important to know that the access mode can change during runtime, eg. based on other features. For example if `TriggerMode` is switched off, the depending features, such as `TriggerSource` might become unavailable. Note that in a GUI representation (eg. in the Simplon Explorer), the temporarily unavailable or locked (read-only) features might be rendered for example as greyed or even disappear. Possible access modes are:

- **Not implemented.** The feature is not implemented. It will stay in this status during the entire operation. Features with this access mode will not be displayed at all in Simplon Explorer GUI.

- **Not available.** The feature is temporarily unavailable (cannot be read nor written). It can, however, become available later, typically based on other features in the same feature tree.
- **Read only.** The feature value is available for reading but not for writing.
- **Write only.** The feature value is available for writing but not for reading.
- **Read-write.** The feature value can be both read and written.

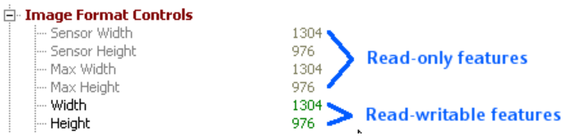


Image Format Controls	
Sensor Width	1304
Sensor Height	976
Max Width	1304
Max Height	976
Width	1304
Height	976

Caching

The feature values are cached internally, which is important especially when working with remote devices, such as GigE Vision cameras, so that the same value does not need to be re-read always over the network. The feature tree knows when it can use the case for every feature and when the cache should be invalidated. The application can rely fully on this mechanism without need to maintain own feature cache — Simplon will communicate with the remote device only when necessary, thus optimizing the performance.

Furthermore, it is useful to distinguish between three types of features from the caching point of view:

- **Write-through caching.** Cached during write operation. This is the case for most features, the exact feature value is used.
- **Write-around caching.** Cached during read operation. Used typically for certain float features of the remote device, when the camera needs to readjust the value to a closest valid one. If the application needs to know the exact used value, it has to re-read the feature value immediately after writing.
- **No caching.** The value is not cached at all. Used for volatile features such as sensor temperature or similar. The application has to start Simplon API polling thread to get the non-cached features updated regularly. Some features might switch to non-cached mode temporarily, for example the gain value will become non-cached when automatic gain mode is active.

Limits

Integer and float features have `minimum`, `maximum` and an `increment` (the increment is optional for floats). The feature value must fall within these limits. Remember that the limits can change during runtime, depending on other features.

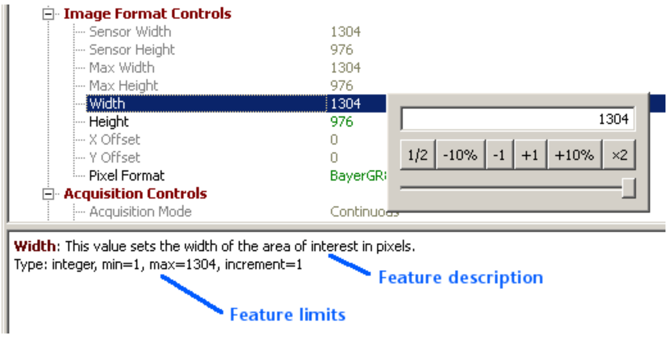


Image Format Controls	
Sensor Width	1304
Sensor Height	976
Max Width	1304
Max Height	976
Width	1304
Height	976
X Offset	0
Y Offset	0
Pixel Format	BayerGR
Acquisition Controls	
Acquisition Mode	Continuo

1304

Width: This value sets the width of the area of interest in pixels.
Type: integer, min=1, max=1304, increment=1

4.1.5.2. Static properties

Unit	String representation of the physical unit represented by the value.
Display format	Provides hint for graphical representation of the feature value, such as whether the feature has linear or logarithmic behavior, if it should be displayed in decimal or hexa form, etc.
Info texts	Each feature gets a `display name` (human readable name of the feature), `tooltip` (short info) and a `description` (longer info). These are useful again especially for display and user interaction.

4.1.6. Feature dependencies

The feature tree knows about possibly complex dependencies between individual features. Changing value of one feature can affect other feature(s) in following ways:

- Change other feature's value — for a feature which is a direct function of the first one.
- Change other feature's limits — eg. increasing a horizontal offset for the image' area of interest decreases maximum of the image width.
- Change other feature's access mode — eg. when disabling a trigger, individual trigger parameters might become unavailable, when starting acquisition, basic acquisition parameters might become locked.
- Change other feature's effective caching mode — eg. gain value is cached for the “manual” mode, but non-cached for the “automatic” gain mode.
- Invalidate the other feature — disabling its cache, for example when resetting the camera to a default status, all feature values must be re-read from the camera rather than from the now-invalid cache.

To understand the feature tree behaviour and treat it properly, it is essential to remember about the possible feature relationships among certain features. The feature tree handles everything for you seamlessly, but you still need to keep in mind that feature status can change based on other features and that for some feature groups the order of writing the features might be significant. To simplify work with such features, Simplon API provides helper functions allowing to set the entire feature group in a single call or even store/reload entire camera configuration.

4.1.7. Chunk data based features

So called “chunk data” are additional data that a camera can deliver to the buffer together with an image. These data can contain information such as image timestamp, frame ID or acquisition parameters used to capture given image. These data can are also part of the feature tree — but they are not read by querying the camera's current status, but rather directly from the buffer itself. The feature tree internal logic knows how to access these additional data in the buffer. Simplon API provides functions allowing to specify the “active” buffer that should be connected to the feature tree.

4.1.8. Event based features

Besides the image and chunk data, the remote device (camera) can also fire asynchronous events. These are useful for variety of purposes, including reporting of asynchronous error and status info, sending log messages and more. The event itself usually caries also additional data describing the event. These data are, yet again, integrated into the feature tree. Simplon API provides notifications about each received event, so that the application can track them and possibly query the data associated with every event.

4.1.9. Feature update notifications in general

The application can instruct the Simplon API to provide notification whenever the status of a given feature changes. The notification would be delivered when feature value has changed, as well as for other status updates (new access mode, feature cache invalidated, etc.).

5. Installation details

This chapter provides detailed overview of Simplon installation process for all supported operating systems. For an express Windows based version (when details are not important for you), please jump instead to Chapter 1, *Quick start* [p. 7].

5.1. Before installation

It is important to know, that the system might not be immediately ready to use after the installation. The hardware (cameras) might need some basic configuration, as well as the system itslef — eg. when using the PicSight-GigE cameras, the network and firewall must be properly configured. The details are always listed in the hardware manual of each respective hardware type. The basics are summarized in Section 1.1, “Connecting and configuring the camera(s)” [p. 7].

5.2. Installation in Windows

5.2.1. System requirements

The Simplon for Windows does not have any particular system requirements, the installer will install all the necessary dependencies, including the .NET Framework runtime. Simplon is compatible with all “recent” Windows versions, including Windows 7, Windows Vista or Windows XP. Note that Windows versions older than Windows 2000 are not supported.

The Simplon package installation should be performed by a user with administrator rights.

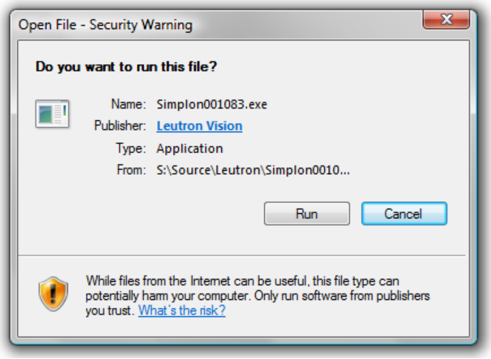
5.2.2. Installation media

The Simplon package can be installed from two types of installation media:

- If you obtained the installation CD-ROM, insert it to the CD-ROM drive. The installation will start automatically.
- If you don't have the installation CD-ROM or if you wish to install a newer Simplon version, you can download the installer from the download area[1] on our website. The installer file name follows pattern `Simplon`*`xyyzzz`*`.exe`, where *`xyyzzz`* stands for Simplon version.

  Execute the installer. Depending on security adjustments of your system and whether you are running the installer from a local disk or a network share, you might get following security warning. Simply proceed clicking Run.

[1] http://www.leutron.com/support-downloads/download-area/



Do you want to run this file?

Name: Simplon001083.exe
Publisher: Leutron Vision
Type: Application
From: S:\Source\Leutron\Simplon0010...

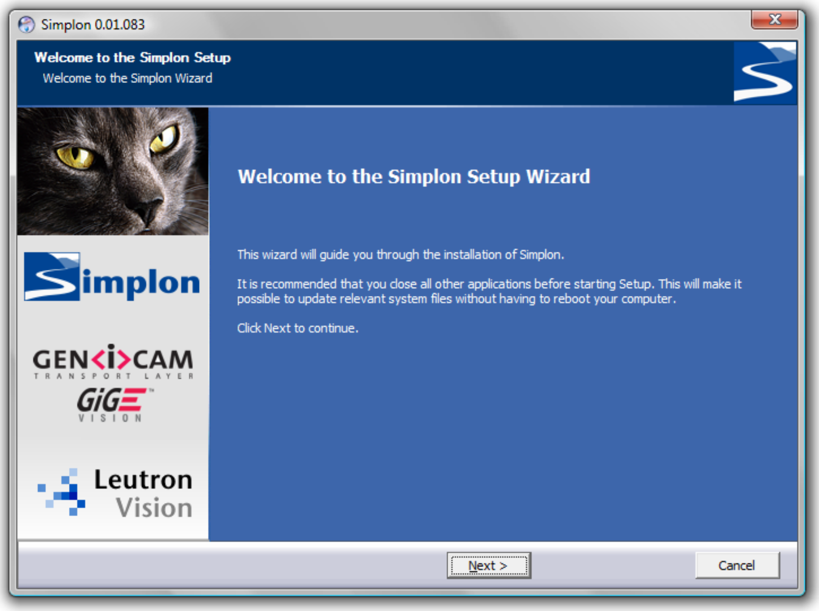
While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

Figure 5.1. Simplon installation: Windows security warning

5.2.3. Installation procedure

The paragraphs below discuss all the dialogs of Simplon's Windows installer and corresponding options. Note that some details might slightly differ between individual Simplon versions, but the principles will always be the same.

The installer first displays a welcome screen with basic information about the software you are going to install. Click Next to proceed.



Welcome to the Simplon Setup
Welcome to the Simplon Wizard

Welcome to the Simplon Setup Wizard

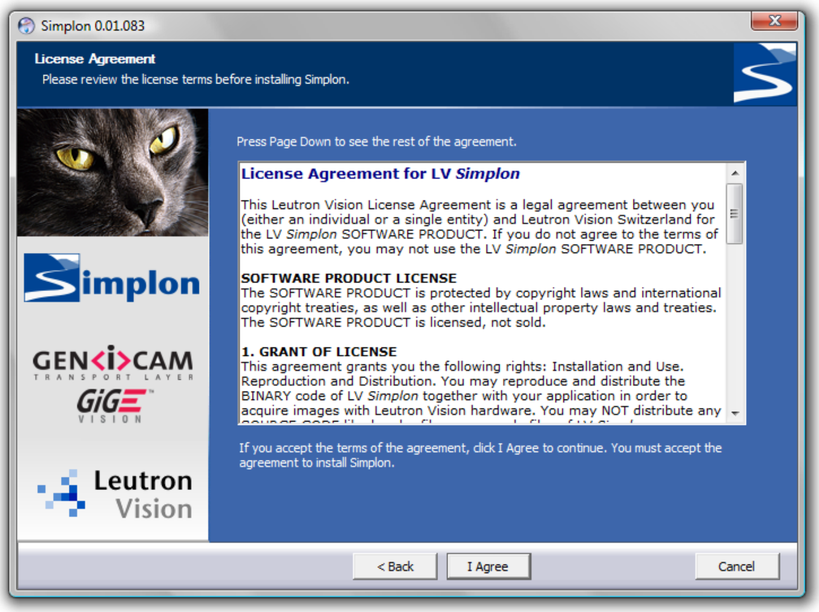
This wizard will guide you through the installation of Simplon.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 5.2. Simplon installation: welcome screen

Next screen is the license agreement. Please read carefully the license text and if you agree, click the I Agree button to proceed with installation. Once confirmed, your full agreement with the license text is assumed.



License Agreement

Please review the license terms before installing Simplon.

Press Page Down to see the rest of the agreement.

License Agreement for LV *Simplon*

This Leutron Vision License Agreement is a legal agreement between you (either an individual or a single entity) and Leutron Vision Switzerland for the LV *Simplon* SOFTWARE PRODUCT. If you do not agree to the terms of this agreement, you may not use the LV *Simplon* SOFTWARE PRODUCT.

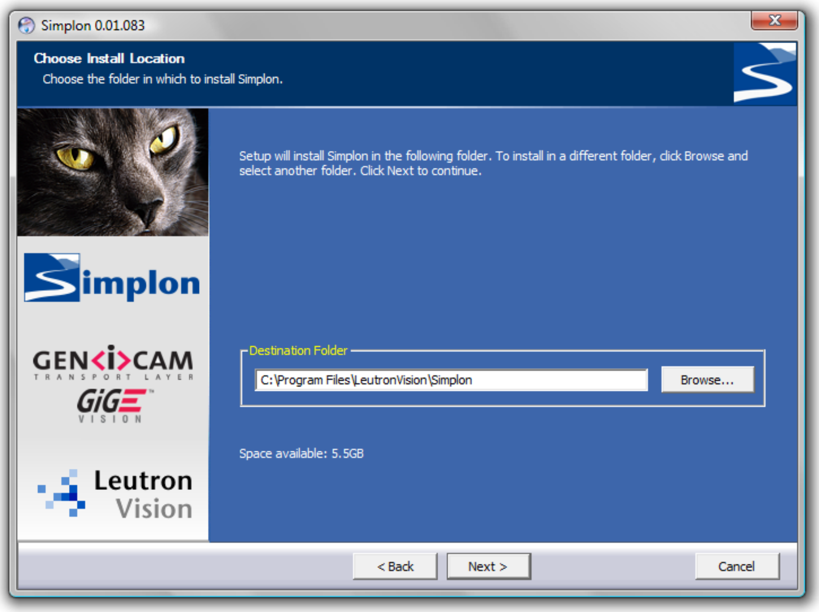
SOFTWARE PRODUCT LICENSE
The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE
This agreement grants you the following rights: Installation and Use. Reproduction and Distribution. You may reproduce and distribute the BINARY code of LV *Simplon* together with your application in order to acquire images with Leutron Vision hardware. You may NOT distribute any

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Simplon.

Figure 5.3. Simplon installation: license agreement

Next the installer asks you about the location where to install Simplon. The default location `C:\Program Files\Leutron Vision\Simplon` should do well in most cases. If you prefer to install Simplon in another directory, select it now. When finished, click the Next button.



Choose Install Location

Choose the folder in which to install Simplon.

Setup will install Simplon in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files\LeutronVision\Simplon

Space available: 5.5GB

Figure 5.4. Simplon installation: destination directory

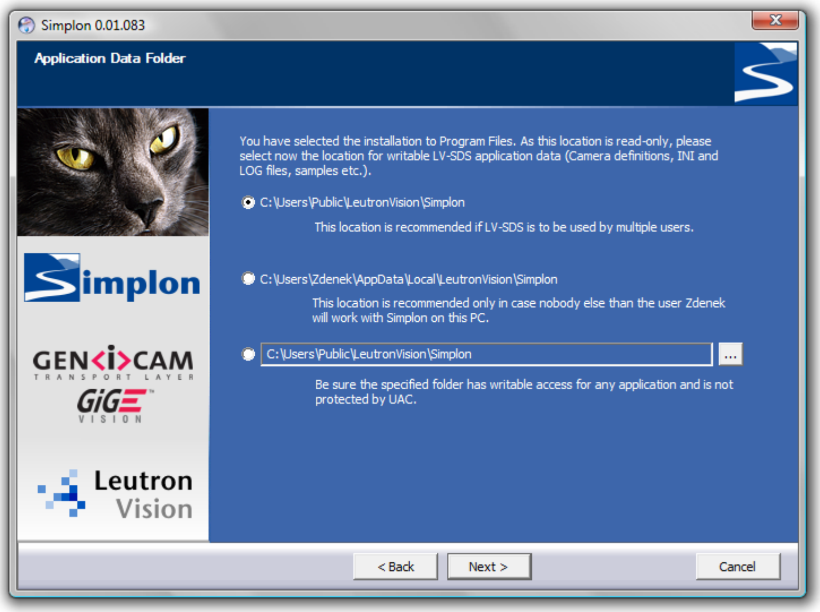
If running Windows Vista or newer Windows version, the default software installation directory (`C:\Program Files`) is read only for regular applications. Simplon, however, requires write access for certain files during operation (Section 5.2.4, “Installation layout” [p. 45]). Therefore the installer will ask you where to put the application data requiring write access.

Note that in Windows versions older than Windows Vista (in particular in Windows XP) the `C:\Program Files` area is not protected and therefore all data can go to a single directory. Also, when installing to a non-standard location (outside `C:\Program Files`), the installer assumes the user chooses a writable *area and installs everything to that directory. The dialog will be omitted in these cases.*

You can choose from three options:

- `C:\Users\Public\LeutronVision\Simplon` — recommended, when Simplon will be used by multiple users.
- `C:\Users\`*`YourUserName`*`\AppDAta\Local\LeutronVision\Simplon` – recommended, when Simplon will be used by a single user (you).
- User configured directory — if none of the options above fit your requirements, you can specify another directory. Be sure the directory has write access for all applications and is not protected by UAC.

Select the preferred option and click Next.



Simplon 0.01.083

Application Data Folder

You have selected the installation to Program Files. As this location is read-only, please select now the location for writable LV-SDS application data (Camera definitions, INI and LOG files, samples etc.).

● C:\Users\Public\LeutronVision\Simplon

This location is recommended if LV-SDS is to be used by multiple users.

○ C:\Users\Zdenek\AppData\Local\LeutronVision\Simplon

This location is recommended only in case nobody else than the user Zdenek will work with Simplon on this PC.

○ C:\Users\Public\LeutronVision\Simplon

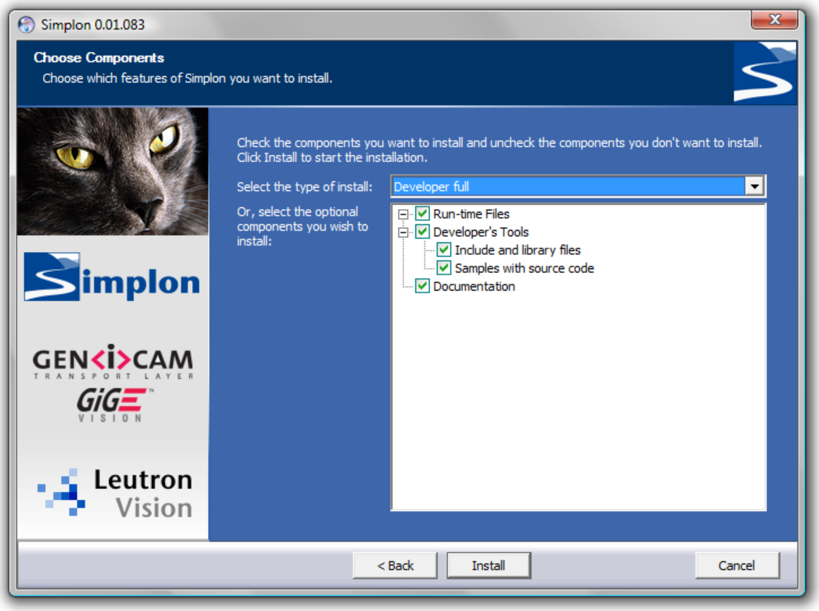
Be sure the specified folder has writable access for any application and is not protected by UAC.

Figure 5.5. Simplon installation: application data directory

In the next step you can select the Simplon components to install. The set of actual options can slightly differ among individual Simplon versions. The basic options are:

- Runtime files — files necessary to run every Simplon application, these are always installed.
- Development files — essential files for compiling against Simplon API as well as additional helper files. Files in this category are typically installed only on a development system, but not deployed with final runtime systems.
  - Include and library files — required for building Simplon API applications.
  - Sample code
- Documentation — full documentation set for Simplon package and all Leutron Vision camera families.

Note that for testing the cameras, the “runtime” components suffice. However, for programming with Simplon you need to make sure the Developer's tools are installed as well. When finished with the selection, click Install.



Simplon 0.01.083

Choose Components

Choose which features of Simplon you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.

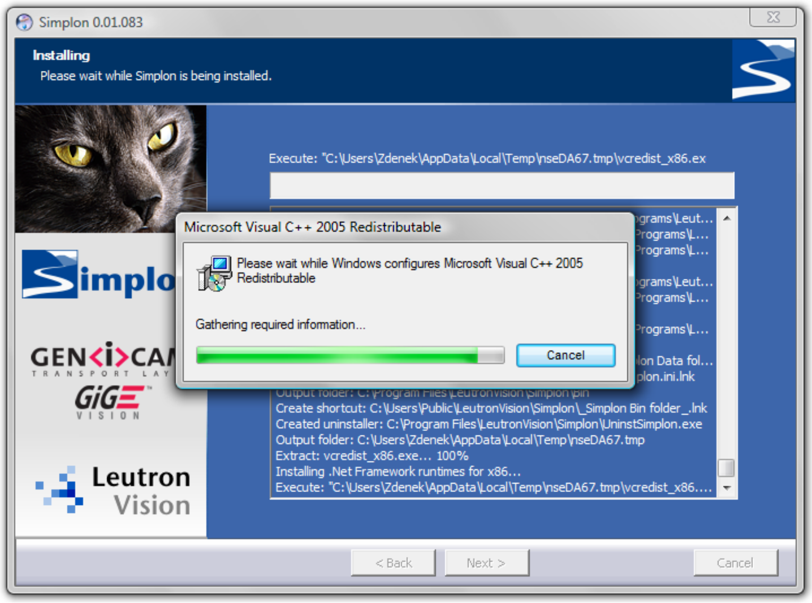
Select the type of install: Developer full

Or, select the optional components you wish to install:

- ☑ Run-time Files
- ☑ Developer's Tools
  - ☑ Include and library files
  - ☑ Samples with source code
- ☑ Documentation

Figure 5.6. Simplon installation: selecting components

After starting the installation, the selected components will be installed to your computer. The installer will inform you about the progress. Note that the installation includes also Microsoft Visual Studio runtime redistributable, device driver for CheckSight cameras and network filter driver for GigE Vision based cameras (PicSight-GigE/PicSight-Smart).



Simplon 0.01.083

Installing

Please wait while Simplon is being installed.

```
Execute: "C:\Users\Zdenek\AppData\Local\Temp\nseDA67.tmp\vcredist_x86.ex
```

```
grams\Leut...
Programs\L...
Programs\L...

grams\Leut...
Programs\L...

Programs\L...

lon Data fol...
plon.ini.lnk
Output folder: C:\Program Files\LeutronVision\Simplon\bin
Create shortcut: C:\Users\Public\LeutronVision\Simplon\_Simplon Bin folder_.lnk
Created uninstaller: C:\Program Files\LeutronVision\Simplon\UninstSimplon.exe
Output folder: C:\Users\Zdenek\AppData\Local\Temp\nseDA67.tmp
Extract: vcredist_x86.exe... 100%
Installing .Net Framework runtimes for x86...
Execute: "C:\Users\Zdenek\AppData\Local\Temp\nseDA67.tmp\vcredist_x86....
```

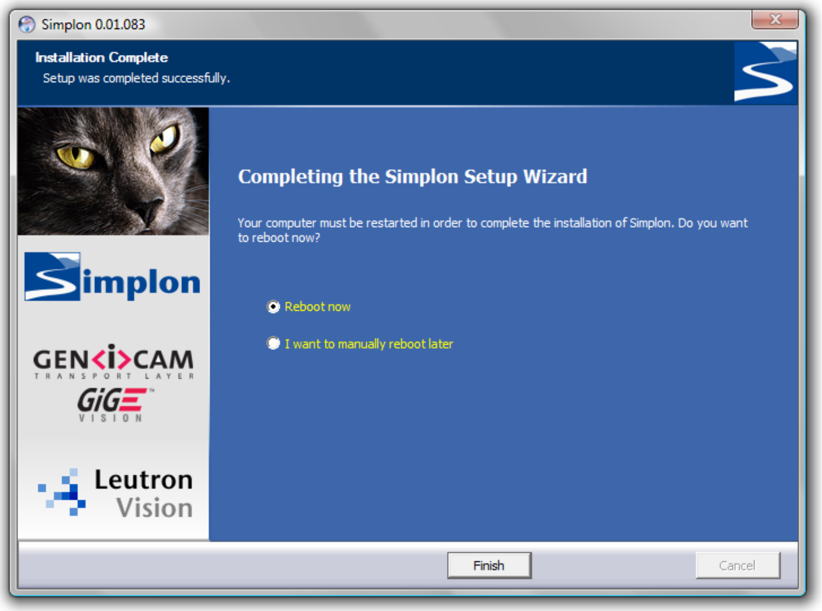
Microsoft Visual C++ 2005 Redistributable

Please wait while Windows configures Microsoft Visual C++ 2005 Redistributable

Gathering required information...

Figure 5.7. Simplon installation: installation progress

When completed, the installer displays the final dialog. We recommend to keep the default option (Reboot now) and click Finish. After rebooting, Simplon is ready to use.



Simplon 0.01.083

Installation Complete

Setup was completed successfully.

Completing the Simplon Setup Wizard

Your computer must be restarted in order to complete the installation of Simplon. Do you want to reboot now?

● Reboot now

○ I want to manually reboot later

Figure 5.8. Simplon installation: finising installation

5.2.4. Installation layout

As mentioned above, the Simplon files are installed to two destinations, the main directory and the application data directory. For the older Windows versions (Windows XP), these two are merged and the application data directory is actually equivalent to the `bin` subdirectory of the main installation destination.

5.2.4.1. Main installation directory

The main installation directory contrains the following subdirectories:

- `backup` — backup of user modified files (such as the configuration files) that might be overwritten when upgrading to newer Simplon version
- `bin` — all executable (`*.exe`) and library (`*.dll` files belonging to Simplon
- `data` — tool-specific data, eg. templates used by the Simplon Source Code Generator
- `doc` — documentation files
- `drivers` — drivers (CheckSight device driver, network filter driver)
- `include` — header files for Simplon API development
- `lib` — library (`*.lib`) files for Simplon API development
- `samples` — additional sample code (most of the samples can be, however, generated interactively using the Simplon Source Code Generator)

5.2.4.2. Application data directory

The application data directory contains following main components:

- `lv.simplon.ini` — Simplon configuration file.
- `lv.simplon.log` — Simplon log file.
- `XML` — contains static XML files copied during installation. None of the files in this directory should be removed unless being explicitly adviced by our support team.
- `XMLCache` — contains GenICam XML files downloaded from cameras.
- `Cache` — preprocessed versions of the GenICam XML files for performance improvement. These can be safely removed, Simplon will re-generate them upon next use.
- `Debug` — the `.pdb` files with debugging information for the Simplon libraries. These might be useful for troubleshooting when contacting our support team.

5.2.5. Installing network filter driver under Windows

The Simplon setup program can install the network filter driver automatically, if desired. The decision can be made on the setup program's dialog window, where installation components are selected. If not installed automatically, it is still possible to install the filter driver manually later, as described below.

Before installing it, you might want to learn more information about the filter driver.

In older Simplon versions the setup did not install the filter driver automatically and the user was required to install it manually.

5.2.5.1. Manual filter driver installation

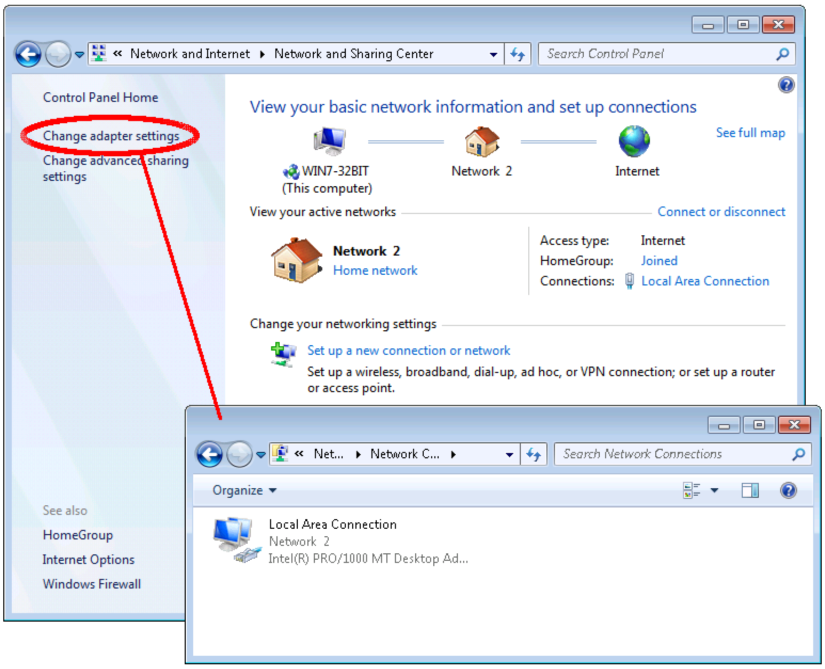
The instructions below describe, how to install the network filter driver manually, if you did not enable it during installation of the Simplon itself. Note that the instructions below assume that Simplon is already successfully installed in the system.

Open the `Network Connections` applet in the Control Panel.

Windows 7 — menu Start → Control Panel → Network and sharing center → Manage Network Connection

Windows Vista — menu Start → Control Panel → Network and internet → Network and sharing center → Manage Network Connection

Windows XP — menu Start → Settings → Control Panel → Network Connections


« Network and Internet ▸ Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
See also
HomeGroup
Internet Options
Windows Firewall
View your basic network information and set up connections
See full map
WIN7-32BIT (This computer)
Network 2
Internet
View your active networks
Connect or disconnect
Network 2
Home network
Access type: Internet
HomeGroup: Joined
Connections: Local Area Connection
Change your networking settings
Set up a new connection or network
« Net... ▸ Network C... ▸
Search Network Connections
Organize ▾
Local Area Connection
Network 2
Intel(R) PRO/1000 MT Desktop Ad...

Set up a wireless, broadband, dial-up, ad hoc, or VPN connection; or set up a router or access point.

Figure 5.9. Windows 7, Installing the network filter driver, step 1

Click with the right mouse button on the network connection, which is used to connect the PicSight-GigE cameras, select Properties from the context menu.

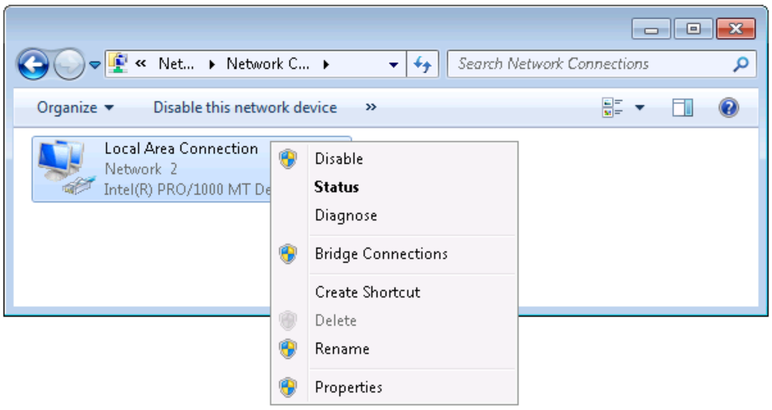
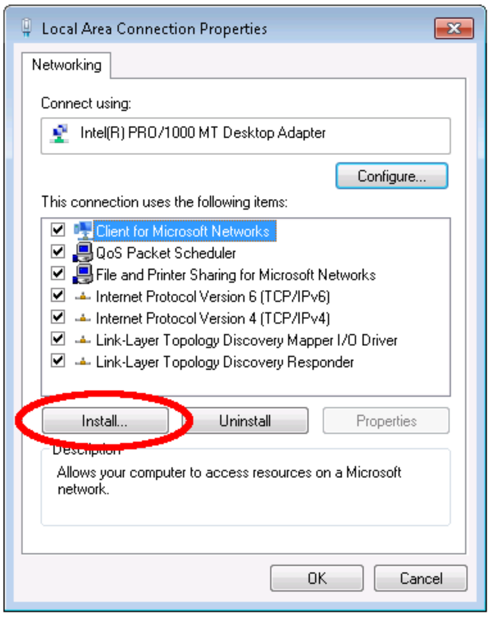


Figure 5.10. Windows 7, Installing the network filter driver, step 2

In the Properties dialog box, General tab select Install... button to start the installation.



Connect using:

Intel(R) PRO/1000 MT Desktop Adapter

This connection uses the following items:

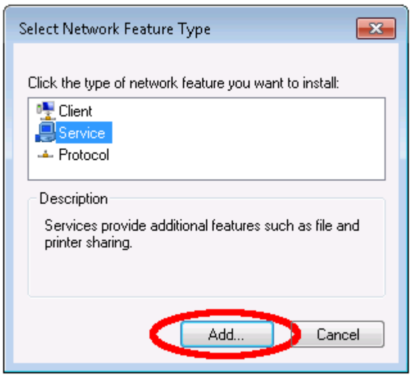
- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Description

Allows your computer to access resources on a Microsoft network.

Figure 5.11. Windows 7, Installing the network filter driver, step 3

The Select Network Component Type dialog box appear. Select Service as the new network component type and click the Add... button.



Click the type of network feature you want to install:

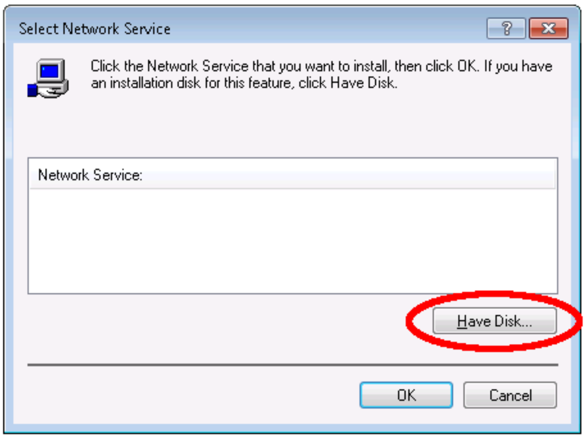
- Client
- Service
- Protocol

Description

Services provide additional features such as file and printer sharing.

Figure 5.12. Windows 7, Installing the network filter driver, step 4

In the Select Network Service dialog box ignore the options possibly offered by the system and click the Have Disk... button to specify the driver location.



Select Network Service

Click the Network Service that you want to install, then click OK. If you have an installation disk for this feature, click Have Disk.

Network Service:

Figure 5.13. Windows 7, Installing network filter driver, step 5

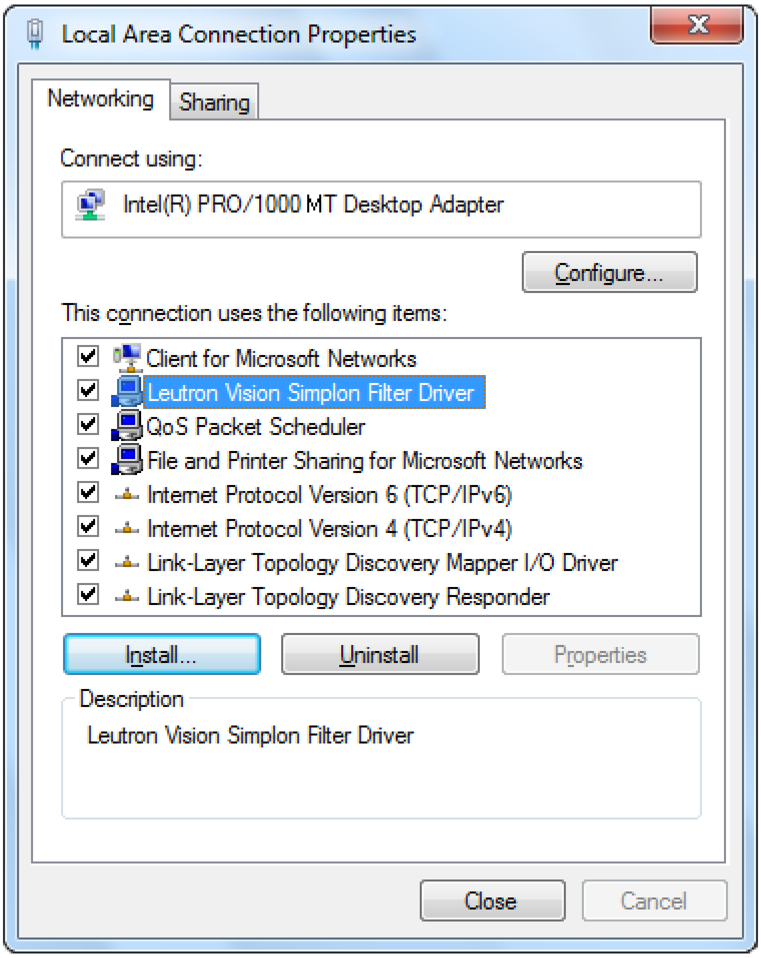
In the next dialog browse to the location where the filter driver files are stored. The directory is `SIMPLONHOME\drivers\Windows32`, where `SIMPLONHOME` is Simplon installation directory, `C:\Program Files\Leutron Vision\Simplon` by default. When located,click Next.

For 64-bit Windows the driver files are located in `SIMPLONHOME\drivers\Windows64`.

The driver consists of 4 files that should in any case be available to the system in the same directory (that you need to locate in the dialog as described above). The files are: `LvGevDrv.sys` (the driver executable itself), the `LvGevDrv.inf` (the file used to install/uninstall the driver), the `LvGevDrv_m.inf` (auxiliary) and the `LvDrv.cat` with a digital signature.

Windows should successfully find the proper driver and offer it for installation as “Leutron Vision Simplon Filter Driver”. Select that option and click the OK button. Depending on Simplon and Windows versions used one or more dialogs might appear to inform you that the driver is not tested for Windows compatibility. In such case, just click through them, allowing to finish the installation successfully.

When finished the filter driver will be successfully installed and ready to use. You should see it listed in the Local Area Connection properties dialog box.


Local Area Connection Properties

Networking | Sharing

Connect using:

Intel(R) PRO/1000 MT Desktop Adapter

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ Leutron Vision Simplon Filter Driver
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Description

Leutron Vision Simplon Filter Driver

Figure 5.14. Windows 7, Installing network filter driver, step 6

5.3. Installation in Linux

This section discusses Simplon package installation under Linux with notes specific for the most popular distributions and overview of system requirements.

5.3.1. Installation package types

Linux world does not have a single standard way for software installation. The Simplon Linux package is therefore distributed in multiple forms to suit all systems.

5.3.1.1. RPM format

- Used on most commercial distributions and their community variants, including Fedora/Red Hat, OpenSUSE/SUSE or Mandriva.
- The package file name is `simplon-x.yy.zzz.rpm`, where `x.yy.zzz` stands for Simplon version.
- To install, execute `rpm -i simplon-x.yy.zzz.rpm` with superuser privileges.
- To uninstall, execute `rpm -e simplon` with superuser privileges.

5.3.1.2. DEB format

- Used on Debian based distributions, including Debian, Ubuntu (and variants) or Mint.

- The package file name is `simplon-x.yy.zzz.deb`, where *`x.yy.zzz`* stands for Simplon version.
- To install, execute `dpkg -i simplon-x.yy.zzz.deb` with superuser privileges.
- To uninstall, execute `dpkg -r simplon` with superuser privileges.

5.3.1.3. Compressed archive (TARGZ)

- The `.tar.gz` archive is intended for systems, where the rpm/deb formats cannot be used or when a customized installation is required for serious reasons. Details are provided in a separate section [p. 61].

5.3.2. System requirements

Following list enumerates the most important requirements for installing and using Simplon. Note that the list might not be complete and is subject to change between Simplon versions.

- Kernel version: kernels of the 2.6 line. The oldest tested version to date is 2.6.18, Simplon is kept aligned with all the newest kernel version.
- Compiler: g++/gcc 4.1 and newer, libstdc++ v. 6 (GLIBCXX 3.4.5), pthreads implementation NPTL.
- When installing on CheckSight, the kernel driver needs to get compiled and configured, therefore the basic infrastructure for kernel development needs to be available, in particular the basic compiling tools (gcc and make) and kernel headers. The name of the package containing the kernel headers might differ among individual distributions (eg. `linux-headers`, `kernel-devel` or similar).

  Additional system requirements might apply for CheckSight systems, if in doubt, contact us to get full list of distributions supported by the CheckSight cameras.
- LSB compatibility and xdg-utils are advantage, but they come by default on most of the usual distributions.
- To compile samples genrated by the Simplon Source Code Generator, one needs basic development tools (make, g++) plus packages for Xlib/Xt development.

5.3.3. Standard installation layout

The Simplon installation follows the Filesystem Hierarchy Standard. Its files are distributed in several directories.

5.3.3.1. Main installation direcotry

The main isntallation directory is `/opt/simplon`.

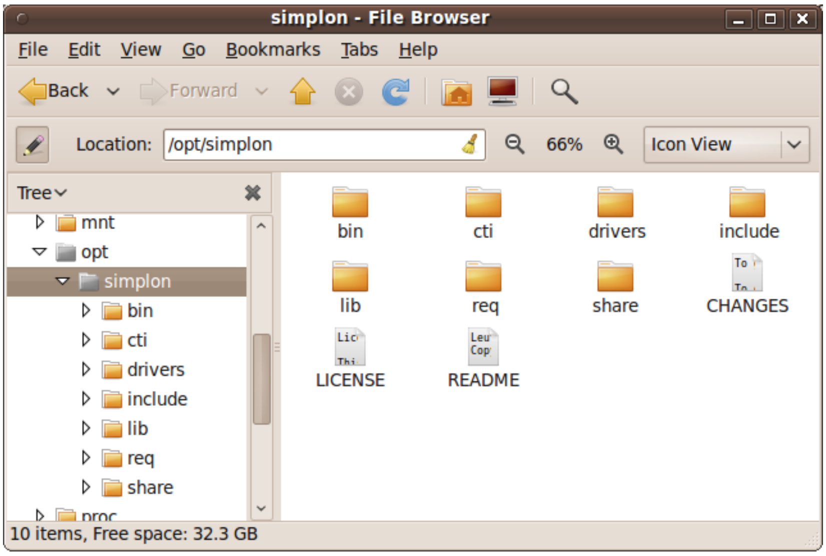


Figure 5.15. Linux installation layout, main directory

The main installation directory contains all the runtime binaries and development files. It is organized in following subdirectories:

- `bin` — tools and test/demo programs, in particular Simplon Explorer (`lv.explorer`) and Simplon Settings (`lv.settings`)
- `lib` — all Simplon API libraries, ie. a directory where the linker should be pointed at; on 64-bit systems the `lib` directory contains the 32-bit versions, while the 64-bit ones reside in `lib64`
- `cti` — contains the Simplon GenTL Producer library; on 64-bit systems the `cti` directory contains the 32-bit versions, while the 64-bit ones reside in `cti64`
- `include` — Simplon API header files
- `share` — static data files used by Simplon
- `share/xml` — static XML files copied during installation
- `req` — foreing requisite components used by Simplon, in particular the GenICam runtime
- `drivers` — scripts and sources necessary to build and install Simplon device drivers (used only on CheckSight systems)

5.3.3.2. Configuration files

The configuration files are stored in `/etc/opt/simplon`.

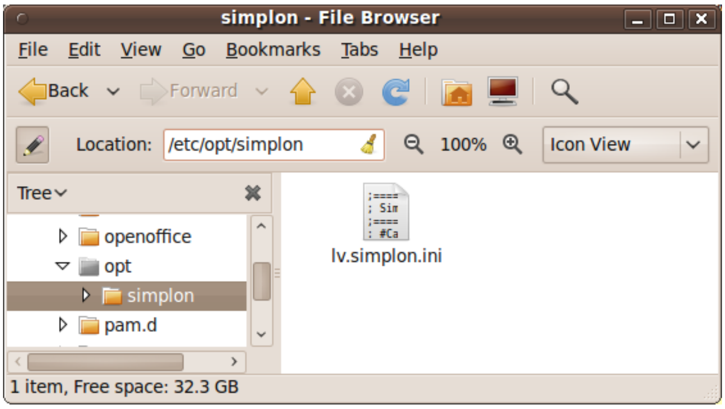
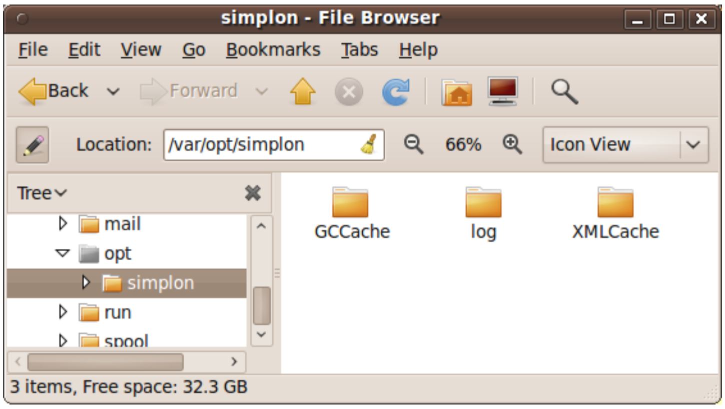

simplon - File Browser
File Edit View Go Bookmarks Tabs Help
Back Forward
Location: /etc/opt/simplon
100%
Icon View
Tree
openoffice
opt
simplon
pam.d
lv.simplon.ini
1 item, Free space: 32.3 GB

Figure 5.16. Linux installation layout, configuration files

The directory contains in particular the main Simplon configuration file, `lv.simplon.ini`. The file should be edited with superuser privileges. See Section 7.1, "Simplon Settings utility and the configuration file" [p. 65] for details.

5.3.3.3. Application data

The various application data, such as logs and cache go to `/var/opt/simplon`.



Location: /var/opt/simplon

3 items, Free space: 32.3 GB

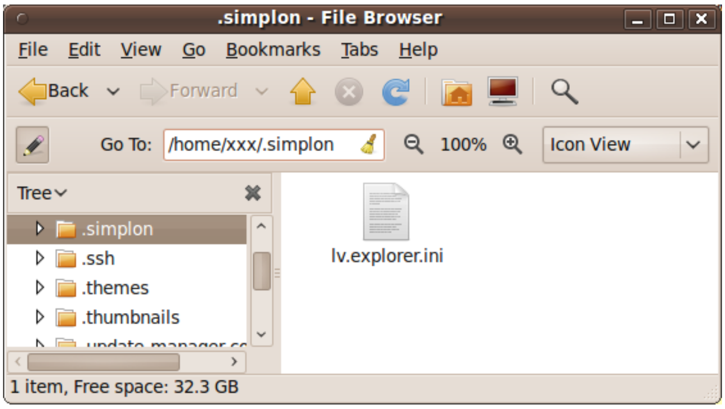
Figure 5.17. Linux installation layout, application data

The data are stored in several subdirectories, in particular:

- `log` — default destination for Simplon log files, in particular the `lv.simplon.log`, that are useful for any troubleshooting (refer to Chapter 9, *Troubleshooting & Support* [p. 70]).
- `XMLCache` — contains GenICam XML files downloaded from cameras.
- `GCCache` — preprocessed versions of the GenICam XML files for performance improvement. These can be safely removed, Simplon will re-generate them upon next use.

5.3.3.4. User specific files

Possible user specific files (such as demos'/tools' configuration) are stored in `~/.simplon`.



Go To: /home/xxx/.simplon

1 item, Free space: 32.3 GB

Figure 5.18. Linux installation layout, user specific data

This directory is used solely to store user specific settings or preferences for Simplon tools, such as Simplon Explorer or Simplon Settings. It will not be needed for usual operation on a runtime system.

5.3.3.5. Other actions performed by the installer

Besides copying the files and adjusting user rights, the installer performs some additional actions, including:

- When installing on a CheckSight system (detected automatically), the necessary device driver is built and installed.
- For convenience, the directory `/opt/simplon/bin` is added to `PATH` environment variable, so that the tools are directly accessible from the shell. This is performed by adding an entry to `/etc/profile.d` — on systems using other mechanism, the system administrator might need to adjust the path manually.
- Directory `/opt/simplon/cti` is added to the `GENICAM_GENTL32_PATH` environment variable (and analogically the 64-bit variant), so that the Simplon GenTL Producer can be located by 3rd party GenTL consumers. This is performed by adding an entry to `/etc/profile.d` — on systems using other mechanism, the system administrator might need to adjust the path manually.
- For convenience, the directory `/opt/simplon/lib` (and it's 64-bit variant) is added to the search path of the dynamic linker. This is performed by adding an entry to `ld.so.conf.d` — on systems using other mechanism, the system administrator might need to adjust the path manually.

5.3.3.6. Custom installations

Installation to other than default locations is possible, however we recommend to do that only for well-legitimate reasons. Corresponding guidelines are provided in a separate section [p. 61].

5.3.4. Notes for the most common distributions

The Simplon package is tested with the most popular distributions. Brief notes for those distributions are listed below. Note that this list by no means suggests that other distributions are not supported, Simplon should work on any system fulfilling the basic requirements (Section 5.3.2, “System requirements” [p. 50]). Also the versions listed for each distribution are just a hint, showing which was the original version used for tests. Each newer version of given distribution should work well and older versions would work if fulfilling the mentioned system requirments.

For each distribution, the tests were done with the most basic “default” setup including X, but without any special additions.

The notes below are related only to Simplon itself, assuming the Linux operating system was properly installed and configured. Special requirements related to operating system installation might apply on CheckSight systems —. To install Simplon on CheckSight, kernel development infrastructure (gcc, make, kernel headers) must be installed to succesfully build and configure the device driver.

5.3.4.1. Ubuntu

- Originally tested version: 9.10 (Karmic Koala)
- Packaging type: deb format
- Works out of the box

5.3.4.2. Debian

- Originally tested version: 5.0 (Lenny)
- Packaging type: deb format
- Works out of the box

5.3.4.3. OpenSUSE (community version of SUSE Linux Enterprise)

- Originally tested version: 11.2

- Packaging type: rpm format
- Default installation switches on a paranoid firewall, which has to be configured (or switched off) if working with GigE Vision cameras (PicSight-GigE/PicSight-Smart)

5.3.4.4. Fedora (community version of Red Hat Enterprise Linux)

- Originally tested version: 12 (Constantine)
- Packaging type: rpm format
- Default installation switches on a paranoid firewall, which has to be configured (or switched off) if working with GigE Vision cameras (PicSight-GigE/PicSight-Smart)

5.3.4.5. Mandriva

- Originally tested version: 2010.0
- Packaging type: rpm format
- Default installation switches on a paranoid firewall, which has to be configured (or switched off) if working with GigE Vision cameras (PicSight-GigE/PicSight-Smart)

5.3.4.6. Slackware

- Originally tested version: 13
- Packaging type: special (tgz based), but slackware has a tool rpm2tgz which can be used to convert the rpm package (`rpm2tgz simlpon-`*`x.yy.zzz`*`.rpm`) and then install it (`installpkg simplon-`*`x.yy.zzz`*`.tgz`)
- Works out of the box

5.3.4.7. Gentoo

- Originally tested version: 10.1
- Packaging type: special, no reasonable way to use the rpm/deb packages, need to use the tar.gz package

5.3.4.8. Mint

- Originally tested version: 8 (Helena)
- Packaging type: deb format
- Works out of the box

5.4. Installation in Windows Embedded

No specific installer for Windows Embedded exists. Simplon should be installed to a running Windows Embedded system using the regular setup program or through its customized version (Chapter 6, *Custom installation* [p. 55]). Windows Embedded component definitions (a `.sld` file) for Simplon are not provided by Leutron Vision, since they would be highly system specific and the system designers should prepare them according their particular needs.

5.4.1. Prerequisites

Components required to install Simplon on a CheckSight system include:

- Windows Image Acquisition Core (group Software/System/Printing & Imaging/Infrastructure/)

- Class Installer - Imaging Devices (group Software/System/System Services/Base/)

6. Custom installation

6.1. Custom installation in Windows

For OEM customers delivering Simplon as a part of their own software package, the “standard” installer can be modified through various options in the `LvSetup.ini` file. These include silent installation, finer control of installed components or disabling specific installer dialog options.

6.1.1. The LvSetup.ini file

In order to enable the integration to the application setup, the Setup can be configured using and `LvSetup.ini` file, placed in the same folder as the Setup file, or at a specified location.

By default, the Setup tries to read the LvSetup.ini file, which is located in the same folder, as the EXE file with the setup. In this INI file various options for the Setup can be defined.

Alternatively, the Setup can read the `LvSetup.ini` file from a different location; this is useful in case the Setup is on a CD-ROM, but you want to make modifications to the `LvSetup.ini` file before the Setup is run (usually by the your application setup program). As modifications cannot be done on a read-only medium, the `LvSetup.ini` file must be located on hard disk.

To force the Setup to read the `LvSetup.ini` file from a different location, the value `SetupIniFile` in the

```
HKLM\Software\Leutron Vision\Simplon\Setup
```

key in Windows Registry must be set to the `LvSetup.ini` file name (with full path), for example:

```
SetupIniFile = "C:\LVSDS\LvSetup.ini"
```

The Setup clears this value after reading it, so it cannot be misused next time the setup is run.

If the `LvSetup.ini` file is missing, or an item is missing in the file, the setup uses a default value, mentioned by each item in this document.

6.1.2. Configuring the Setup

The `LvSetup.ini` file has the following structure:

```
[General]
SilentMode=0
ReadMeFile=
LicenseFile=
InstallPath=
InstallUserPath=
ForceInstallPath=0
ToProgramFiles=1
MenuShortcuts=1
DisableFileLog=0
Force32BitInstall=0
Include32In64=0
InstallCRuntime=1
InstallKernelDrivers=1
InstallFilterDrivers=1
InstallPdbFiles=1

[Dialogs]
```

```
DisplayWelcome=1
DisplayReadme=0
DisplayLicense=1
AskForPath=1
AskForComponents=1
DisplayFinish=1
EnableReboot=1

[Documentation]
Install=1

[IncludeLib]
Install=1

[Samples]
Install=1
```

6.1.3. [General]

6.1.3.1. SilentMode

1 — the Setup runs without asking the user anything, even if the other options, like `DisplayWelcome` or `AskForPath`, are set to 1. If you use this option, be sure you specify the `InstallPath` correctly.

0 — the setup runs normally.

Default value: 0

6.1.3.2. ReadMeFile

If not specified, the Setup does not display any Read Me dialog.

If specified, the Setup reads the contents of the file and displays it. The ReadMe file can be either a plain text (with line breaks at the end of paragraphs, not each line), or it can be an RTF file with text formatting.

If you specify the name of the file without the path, like

```
ReadMeFile=readme.rtf
```

then the file is expected to be at the same location, as the Setup file.

Alternatively, you can specify the file with its full path, for example:

```
ReadMeFile=C:\Data\MySimplonSetup\readme.txt
```

Default value: empty (no dialog displayed)

6.1.3.3. LicenseFile

If not specified, the Setup displays standard License dialog.

If specified, the Setup reads the contents of the file and displays it instead of the standard message. The License file can be either a plain text (with line breaks at the end of paragraphs, not each line), or it can be an RTF file with text formatting.

If you specify the name of the file without the path, like

```
LicenseFile=license.rtf
```

then the file is expected to be at the same location, as the Setup file.

Alternatively, you can specify the file with its full path, for example:

```
ReadMeFile=C:\Data\MySimplonSetup\license.txt
```

Default value: empty (standard license text displayed)

6.1.3.4. InstallPath

The name of the folder, to which the libraries are to be installed. If specified without path, like

```
InstallPath=Simplon
```

the Setup uses (in this example) either "C:\Simplon" or "C:\Program Files\Simplon" folder for the installation, depending on the `ToProgramFiles` value (see below).

If specified with full path, like

```
InstallPath=C:\Prg\Simplon
```

the Setup uses this path, regardless of the `ToProgramFiles` value.

Important note: If Simplon files are already found to be installed on the PC, the `InstallPath` may be ignored - see the `ForceInstallPath` option below.

Leave this value empty if you want to use the default installation path.

6.1.3.5. InstallUserPath

The name of the folder, to which the *writable* files are to be installed. If specified without path, like

```
InstallPath=Simplon
```

then the Setup uses the public user folder for the installation.

If specified with full path, like

```
InstallPath=C:\Users\Public\Simplon
```

the Setup uses this path without modifications.

Important note: If Simplon files are already found to be installed on the PC, the `InstallUserPath` may be ignored - see the `ForceInstallPath` option below.

Leave this value empty if you want to use the default user path.

This option is used only for Windows Vista and higher. On Windows XP the installation folder is considered to be writable and all the files are installed there.

6.1.3.6. ForceInstallPath

0 — in case the Simplon files are already installed on the PC, the Setup program uses for installation the path, on which it found the previous installation, instead of the path specified in the `InstallPath` and `InstallUserPath`.

1 — if you want to force the setup to use the `InstallPath` and `InstallUserPath` values at any case, set `ForceInstallPath` to 1.

Default value: 0

6.1.3.7. ToProgramFiles

1 — the Setup adds the Program Files folder path before the `InstallPath`, so for example, if the `InstallPath` is set to `Simplon`, the target folder will be "`C:\Program Files\LeutronVision\Simplon`". This applies only in case the `InstallPath` was not already specified with the full path.

0 — only the drive letter of the system disk is added: "`C:\Simplon`".

In case you prefer installation to a subfolder of the Program Files folder, use this option to do so, instead of direct setting the "`C:\Program Files\...`" in the `InstallPath` - keep in mind that the default Program Files folder need not necessarily be on the C: drive and need not be named “`Program files`” in all versions of Windows (for example on x64 Windows the Program files folder has the "(x86)" attached, on same other language versions the Program Files are translated etc.)

Default value: 1

6.1.3.8. MenuShortcuts

1 — the menu shortcuts are added to the Start Menu, under `Programs->Leutron Vision->Simplon`

0 — the menu shortcuts are not created.

Default value: 1 for full installation, 0 for run-time installation.

6.1.3.9. Include32In64

0 — on x64 Windows the setup will install only x64 components

1 — on x64 Windows the setup will install both x32 and x64 components. The components which need QT libraries will function only in 64-bit version as the QT libraries cannot coexist in both 32-bit and 64-bit versions.

Default value: 0

6.1.3.10. DisableFileLog

1 — The setup will after installation write to the `lv.simplon.ini` file the `Enable=0` in the section `[Log.File]` — this disables writing the log to the file (useful on systems where writing to a log file is not desirable, for example on a system with a flash memory instead of hard disk).

0 - The setting in `lv.simplon.ini` will not be touched.

Default value: 0

6.1.3.11. InstallCRuntime

1 - Installs the C/C++ runtime libraries, needed for Simplon DLLs.

0 - Skips the installation of the runtimes. If the runtimes are not present on the PC, Simplon will not be usable.

Default value: 1

6.1.3.12. InstallKernelDrivers

Installs or preinstalls the kernel driver, needed for Leutron PCI hardware, like the CheckSight camera module or a framegrabber.

1 — modifies the component menu that the kernel driver is by default installed.

0 — modifies the component menu that the kernel driver is by default not installed

FORCE — the kernel driver is installed, and the user cannot change this option.

DISABLE — the kernel driver is not installed and the user cannot change this option.

Default value: 1

If there is a Leutron Vision hardware found on the PC, the choice is automatically switched to FORCE.

Can be used on Windows Vista and newer. On Windows XP the driver installation must be done manually, using the New Hardware Found wizard; then the Simplon setup only updates existing installed driver.

6.1.3.13. InstallFilterDrivers

Installs or uninstalls the filter driver, needed for GigE cameras.

1 — modifies the component menu that the filter driver is by default installed.

0 — modifies the component menu that the filter driver is by default not installed.

FORCE — the filter driver is installed, and the user cannot change this option.

DISABLE — the filter driver is not installed and the user cannot change this option.

Default value: 1

If the choice is checked, the driver is installed or updated, if the choice is not checked and the driver is installed from the past, it is uninstalled.

6.1.3.14. InstallPdbFiles

The PDB files are helper files for debugging the application. If the debugger has the PDB files, it can show more information about the debugged code. The Simplon PDB files are usually necessary only when you need to debug some unexpected program or library behavior, like a crash.

1 — modifies the component menu that the PDB files are by default installed.

0 — modifies the component menu that the PDB files are by default not installed

FORCE — PDB files are installed, and the user cannot change this option.

DISABLE — the PDB files are not installed and the user cannot change this option.

Default value: 1

6.1.4. [Dialogs]

6.1.4.1. DisplayWelcome

1 — the Welcome dialog box is displayed, when Setup starts.

0— the Welcome dialog box is skipped.

Default value: 1

6.1.4.2. DisplayReadme

1 — if the ReadMe file was specified by the `ReadMeFile` item, the ReadMe dialog box is displayed during the setup.

0— the ReadMe dialog box is skipped.

Default value: 1

6.1.4.3. DisplayLicense

1 — the License dialog box is displayed during the setup.

0— the License dialog box is skipped.

Default value: 1

6.1.4.4. AskForPath

1 — the dialog box for confirming or selecting the install path is displayed.

0 — the install path cannot be specified by the user and the installation goes to the predefined installation folder, see `InstalPath` items.

Default value: 1

6.1.4.5. AskForComponents

1 — the dialog box with the Components selection is displayed, so the user can modify the selection of the components.

0 — the user cannot select the components and the ones predefined in the INI file are installed.

Default value: 1

6.1.4.6. DisplayFinish

1 — the Finished dialog box is displayed when the Setup completes

0 — the Finished dialog box is skipped when the Setup completes

Default value: 1

6.1.4.7. EnableReboot

The Setup requires the PC to be rebooted after the Setup completion. Before rebooting, this dialog box is displayed

However, if you are running the Setup from another setup, you might want to avoid the reboot and do it after the whole setup of your application is finished. By setting `EnableReboot` to 0, the prompt for rebooting at the end of the Setup is removed and the Setup does not require reboot.

Keep in mind that Simplon components may not be functional until the reboot is actually done.

Default value: 1

6.1.5. [Documentation]

6.1.5.1. Install

1 — modify the component menu that the manuals are by default installed.

0 — modify the component menu that the manuals are by default not installed

FORCE — manuals are installed, and the user cannot change this option.

DISABLE — the manuals are not installed and the user cannot change this option.

Default value: 1

6.1.6. [IncludeLib]

6.1.6.1. Install

1 — modify the component menu that the *include* and *lib* files are by default installed.

0 — modify the component menu that the *include* and *lib* files are by default not installed

FORCE — the *include* and *lib* files are installed and the user cannot change this option.

DISABLE — the *include* and *lib* files are not installed and the user cannot change this option.

Default value: 1

6.1.7. [Samples]

6.1.7.1. Install

1 — modify the component menu that the samples are by default installed.

0 — modify the component menu that the samples are by default not installed.

FORCE — the samples are installed and the user cannot change this option.

DISABLE — the samples are not installed and the user cannot change this option.

Default value: 1

6.2. Custom installation in Linux

In some situations, the official Simplon installers (RPM or DEB) cannot be used, either because given system does not properly support either of them or when installing to other than default locations is required for serious reasons. To cover those situations, the Simplon package can be on request delivered in the `.tar.gz` format. This allows full flexibility during the software installation, however, certain rules need to be followed to make the installation fully functional. Note that the installation details are subject to change — in case of doubts or experiencing a problems with the installation, contact our support team. It is highly recommended to use the standard installers wherever possible.

Before planning the installation, you might want to read the section describing the regular installation, including the system requirements and the default installation layout: Section 5.3, “Installation in Linux” [p. 49].

6.2.1. Extracting the archive

The package file name is `simplon-x.yy.zzz.tar.gz`, where *x.yy.zzz* stands for Simplon version. As a first step, the files from the archive should be extracted to the system:

- The `opt` directory should be extracted to `/opt`. Alternatively, it can be extracted to a custom location, provided that the environment is properly adjusted, see below.
- The `etc` directory should be extracted to `/etc`. Alternatively, it can be extracted to a custom location, provided that the environment is properly adjusted, see below.
- The `var` directory (if present in the archive) should be extracted to `/var`. Alternatively, it can be extracted to a custom location, provided that the environment is properly adjusted, see below. If the directory is not present in the archive, we recommend to create it in the target system, including (empty) subdirectories `log`, `XMLCache` and `GCCache`.

6.2.2. Adjusting the environment

When installing the Simplon files to other than default locations, Simplon must be informed about the custom installation locations through environment variables to work properly:

- `LVS_ROOT` — should point to the main installation directory (replacement of `/opt/simplon`).
- `LVS_CFG` — should point to the configuration files directory (replacement of `/etc/opt/simplon`).
- `LVS_VAR` — should point to the application data directory (replacement of `/var/opt/simplon`).
- `LVS_USER` — should point to the directory for user configuration files (replacement of `~/.simplon`).

It is of course possible that more of them point to the same location. The system administrator is responsible for adjusting correct access rights — in particular the user configuration and application data areas should be writeable.

The path to the Simplon GenTL Producer needs to be adjusted according to the GenTL standard requirements:

- On 32-bit installations, the variable `GENICAM_GENTL32_PATH` must point to the `cti` subdirectory of the main Simplon installation.
- On 64-bit installations, the variable `GENICAM_GENTL64_PATH` must point to the `cti64` subdirectory of the main Simplon installation.

On systems supporting the `/etc/profile.d` mechanism, this will be handled automatically by the file `etc/profile.d/simplon.sh` in the `.tar.gz` archive. The file, however, assumes default installation locations — in case of custom installation, its contents needs to be adjusted accordingly.

The `PATH` variable could be adjusted to include the `bin` subdirectory of the main Simplon installation directory, so that the individual Simplon applications can be loaded from the console without specifying the full path.

6.2.3. Dynamic linking

If desired, the directory with the Simplon shared libraries should be added to the search path of the dynamic linker. The shared libraries are located in `lib` or `lib64` (depending on architecture) subdirectory of the main Simplon installation.

On systems supporting the `/etc/ld.so.conf.d` mechanism, this will be handled automatically by the file `etc/ld.so.conf.d/simplon.conf` in the `.tar.gz` archive. The file, however, assumes default installation locations — in case of custom installation, its contents needs to be adjusted accordingly. To apply the configuration, the `ldconfig` command will have to be executed.

The dependencies of the individual Simplon libraries and executables should be resolved automatically — the corresponding links are stored in given files using the DT_RPATH/DT_RUNPATH ELF tags. The tags contain both the absolute path (pointing to the default installation locations) and the relative path (using the "$ORIGIN" convention). This means that dependencies should be automatically resolved even when installing to a custom location.

Note that in certain situations (such as when using suid binaries), the $ORIGIN based path might be ignored by the dynamic linker. In such situation, you might need to adjust the runpaths to match your installation, or use another equivalent mechanism (such as `LD_LIBRARY_PATH`) of your choice.

6.2.4. Access rights

- The execution rights for all the executables should be adjusted according to your system policy. The default installers enable the execution rights for all users.

  Additionally, you might want to add the setuid bit for the `lv.fwupgrade` so that the tool is allowed to automatically shutdown the CheckSight system after firmware update.

- The `/var/opt/simplon` directory (or its equivalent specified through `LVS_VAR` environment variable) and its subdirectories must have write access for all users.

- All other installed files should have at least read access for all users enabled.

6.2.5. Kernel drivers

Depending on the hardware you are going to use, you might need to install, build and configure the kernel driver(s) delivered with Simplon. This applies in particular to the CheckSight device driver (not needed when using only PicSight cameras), but in future Simplon versions other drivers might be available.

The driver(s) are available in the drivers subdirectory of the main Simplon installation. There's one subdirectory per driver. Each driver comes with a control script (typically named `control_driver`). This script is used by the default installers to build, install and configure the driver. The script is easy to follow and commented, so it should be easy for the system administrator to understand its purpose and eventually adapt the script to match the desired target system. Note that the script is subject to change any time between Simplon versions without prior notice.

6.2.6. Optional files

Some of the files distributed with the Simplon package might not be necessary for the basic runtime operation.

- The tools installed in the `bin` subdirectory are very useful in the development phase, but might not be necessary in the target runtime installation. In such case they can be freely omitted without any harm.

- The 3rd party prerequisite libraries, which are not typically installed on every system, or might be present in different version, are distributed together with Simplon in the `req` subdirectory. This covers especially GenApi, Qt, Xerces/Xalan. On embedded, space-sensitive installations, where the system already contains one or more of these components, the "original" copies can be used instead of those delivered in the Simplon package. It is, however, fully responsibility of the system administrator to determine and use the proper versions of these prerequisites for every given Simplon version and configure the dynamic linking properly. Do this only at your own risk.

- The support files for the Simplon Source Code Generator (in `share/srcgen`) can be omitted if Simplon Source Code Generator functionality is not needed.
- The contents of the `drivers` directory might or might not be needed, depending on the hardware you are going to use. It is fully responsibility of the system administrator to install and configure all the required drivers.
- The configuration files from the `etc` tree can be omitted if you compensate their purpose by other means (most of them were discussed above). The only exception is the `lv.simplon.ini` file, which must always be present.

6.2.7. Uninstallation/upgrade

When uninstalling or upgrading the Simplon package, all relevant steps corresponding with your custom installation scenario should be performed to keep the system consistent.

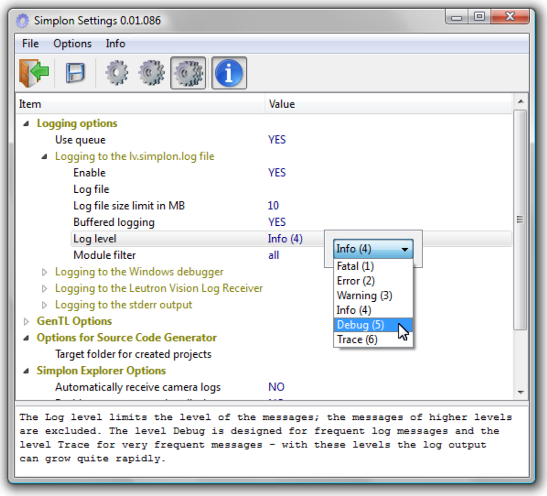
7. Configuration

7.1. Simplon Settings utility and the configuration file

The Simplon configuration options are concentrated in a single file, `lv.simplon.ini`. On Windows systems the file is located in the “application data” part of the installation. On Linux systems it's stored by default in `/etc/opt/simplon`.

The file can be edited directly — every single option is well documented directly in the file, including it's possible values. Note that on Linux superuser privileges are required to edit the file.

Instead of editing the file directly, however, the recommended alternative is to edit the file through a helper utility, Simplon Settings (Windows: menu Start → Programs → Leutron Vision Simplon → Tools → Simplon Settings; Linux: /opt/simplon/bin/lv.settings as superuser). The tool provides a comfortable way to edit the `lv.simplon.ini` file.



Item	Value
Logging options	
Use queue	YES
Logging to the lv.simplon.log file	
Enable	YES
Log file	
Log file size limit in MB	10
Buffered logging	YES
Log level	Info (4)
Module filter	all
Logging to the Windows debugger	
Logging to the Leutron Vision Log Receiver	
Logging to the stderr output	
GenTL Options	
Options for Source Code Generator	
Target folder for created projects	
Simplon Explorer Options	
Automatically receive camera logs	NO

```
The Log level limits the level of the messages; the messages of higher levels
are excluded. The level Debug is designed for frequent log messages and the
level Trace for very frequent messages - with these levels the log output
can grow quite rapidly.
```

Figure 7.1. Simplon Settings utility

The tool's interface can be controlled threw a handful of menu options (or corresponding buttons). You can show/hide the info pane with info about the selected configuration option. You can switch between three configuration levels (beginner, expert, guru) — the higher level, the more detailed options are shown. Finally, you can store the changes.

The set of options itself is ever evolving, so the Simplon Settings tool itself (or the `lv.simplon.ini` file) is the most complete documentation. The options include detailed logging configuration (see also Chapter 9, *Troubleshooting & Support* [p. 70]), system installation related info and configuration of individual Simplon subsystems.

7.2. Selected configuration options

The documentation for all the Simplon configuration options is provided directly in the Simplon Settings utility and the configuration file. Here we'll mention just few selected important options:

Logging options

The Simplon log is an important tool for any support and troubleshooting. For regular operation, the log might be kept at low level (default is level `Info (4)`) or even switched off to optimize the performance. However, during application development or when debugging a problem, we recommend to switch the `Log level` configuration option to `Trace (6)`. All log files sent to our support team should be created with level `Trace (6)`.

The logging infrastructure is optimized for high performance. For example, the logging to file is buffered to minimize the disk access overhead. This can have a drawback, however, that in case of an application crash, the last part of the log might be missing in the file. When troubleshooting application crashes, we recommend to switch on the option `Disable all buffering` .

Heartbeat timeout

`Heartbeat timeout` is an important parameter, related to GigE Vision cameras only (PicSight-GigE/PicSight-Smart).

This timeout, in milliseconds, defines how frequently has the camera send the “heartbeat” messages to the camera, indicating that it's still alive. When the camera does not receive the message within the specified timeout period, it assumes the application has gone away and disconnects itself. The default value used in Simplon is 3 s.

At most times, the heartbeat handling is fully transparent to the user. However, during debugging, when a developer stops the application at a breakpoint for single stepping, the Simplon heartbeat handling thread would also stop, fail to deliver the heartbeat messages and the camera would disconnect. To prevent that, we recommend to increase the heartbeat timeout parameter to a higher value. The exact value should be a compromise between being able to debug comfortably, while understanding that when the application stops unexpectedly without disconnecting the camera properly, it will take a while before the camera disconnects automatically and becomes available for another connection.

The parameter is configurable through camera features at runtime, while its default value can be adjusted through the Simplon Settings utility.

8. Vision Standards

Following paragraphs provide a brief information about the most important machine vision standards built into Simplon and the PicSight/CheckSight camera families.

8.1. GenICam

The main idea of the GenICam standard, published first in 2006 under EMVA (European Machine Vision Association), is to provide a unified application programming interface (API) to the users of machine vision cameras. It enables an easy integration of individual components, such as cameras, image processing libraries, drivers or frame grabbers.

GenICam is independent on the transport layer technology. The GigE Vision standard uses GenICam to access camera configuration. GenICam is also implemented by manufacturers of cameras based on "older" interface technologies (Camera Link, USB, IIDC DCAM) and efforts are made to back-integrate GenICam into these standards. More importantly, newly developed machine vision standards are expected to build upon GenICam and smart camera platforms, such as Leutron Vision CheckSight also heavily rely on GenICam based interoperability. GenICam is thus supported by virtually all important manufacturers of machine vision components.

GenICam consists of three main modules, GenApi, GenTL, and SFNC.

8.1.1. GenICam GenApi

GenApi (GenICam Application Programming Interface) is a basic building block of GenICam. It allows to describe complete camera functionality using an XML file with precisely defined syntax. Individual camera features are described by their name, type (integer, float, boolean, command, string, etc.), address, length and other parameters of the register controlling the feature, as well as other details including complex description of logical and mathematical relationships between individual features. The XML file is usually stored directly on the camera, which further simplifies the automatic configuration process.

The standard clearly defines the way, how the XML configuration file should be interpreted, so that the host-side implementation (application itself) and camera cooperate in harmony. GenICam also provides quality reference implementation in form of a set of C++ libraries with license similar to BSD. All GenICam software products known to the author are using the reference implementation.

GenApi is fully independent on the transport layer, which only has to supply two functions for reading and writing the camera registers by means of its corresponding protocol (GVCP in case of GigE Vision).

The GenApi principle is revolutionary among other aspects in the degree of flexibility available for the camera description. Former standards either did not define details of the communication protocol at all (Camera Link), or defined the protocol, but bound to a rigid register layout (IIDC DCAM), thus disabling any reasonable extensibility. GenICam GenApi allows exact camera description, while leaving full freedom for layout and implementation of individual registers, as well as for defining custom features, not specified by the standard.

8.1.2. GenICam GenTL

GenTL (Transport Layer) is the newest addition to GenICam. It defines interface for acquisition of image sequences (or additional non-image data) independently on the transport layer technology and platform (operating system, programming language, etc.).

GenTL allows to enumerate and identify the devices (cameras) connected to the system, control access to them from individual applications (“GenTL Consumers”), configure them (by means of GenApi), configure the pool of buffers used for the acquired data and control the acquisition itself, including synchronization, buffer locking, control of the input and output buffer queues, reading additional data bound to individual images, etc.

The basic camera functionality is available through programming interface compatible with the ISO norm for the C programming language. More advanced GenTL features can be configured by means of a GenApi interface, similarly as the camera itself. In such case the GenTL Producer implements a virtual register space, described by a similar XML file, which is used also for description of the cameras.

Thanks to GenTL, the applications can use cameras across the transport layer technologies, without knowing any low-level details about each respective technology.

8.1.3. GenICam SFNC

SFNC (Standard Features Naming Convention) is kind of superstructure on top of GenApi. It defines a convention for naming features typical for machine vision cameras. SFNC defines an universal camera model, which might be implemented by most of the cameras on the market. Generic applications and libraries can, thanks to this standard model, automatically comprehend virtually any newly connected camera and configure it according it needs, without "manual" intervention from the programmer or user.

SFNCC thus practically takes the full freedom of GenApi and ties it again to a clearly defined model, so the features listed in the XML configuration file can be automatically (ie. without study of the documentation) bound to a particular meanings. Nevertheless, the camera implements this model only to the level possible and practical for its design. The designer has still full freedom for adding his own extensions and features not defined by the standard.

The SFNC model is very comprehensive, allowing to define most of the camera functionality, perhaps just with exception of very specific features.

8.2. GigE Vision

GigE Vision is relatively young but already very mature and widely adopted standard published in 2006 under AIA (Advanced Imaging Association). It defines interface (set of communication protocols) for industrial cameras using Ethernet as its transport media. The goal of the standard was to reach high degree of interoperability between individual manufacturers. Despite of the name, suggesting that it was originally designed for gigabit Ethernet, the standard itself does not define any particular transport media and enables thus transition to faster media (10G Ethernet) in future.

Main advantages of GigE Vision against competing technologies are:

- Gigabit Ethernet infrastructure is readily available in most current systems and is well known.
- Sufficient bandwidth for most of current camera designs. Larger bandwidth is offered only by Camera Link, but after arrival of the 10G Ethernet even this will not be true.
- Cable length up to 100 meters without regeneration, with switches the length is practically unlimited.
- Number of cameras available in the system is practically unlimited.
- Wide spectrum of network topologies, support for multicasting etc. enable to build specific systems with eg. multiple cameras simultaneously available (and simultaneously acquiring) to multiple host computers.
- Plug&play, support of GenICam standard.
- Cost savings thanks to utilization of cheap, off-the-shelf equipment such as network cards (instead of specialized frame grabbers), standard network cables, etc. Total costs can compete even with analog systems.

GigE Vision is actually set of protocols built on top of UDP/IP. UDP was selected to achieve maximum possible performance. Error checking and repairing mechanisms are available directly by means of the GigE Vision protocols, if it is required by the application.

Main components of the standard are Device Discovery, GVCP, GVSP, Bootstrap registers:

Device discovery

Mechanism allowing discovery and identification of GigE Vision compatible cameras available in the system. The application issues (broadcast) a request for identification of the cameras. Individual cameras must react to this request by sending their card, containing for example the camera's IP address, name of the camera manufacturer and model name, as well as other basic information.

The application can change the IP configuration of the camera (eg. to bring it to its own subnet) and request exclusive rights to access the camera.

GVCP

Once the connection is established, the application communicates with the camera using GVCP (GigE Vision Control Protocol). Its task is to control all the communication with the camera, especially read and write all the camera's control registers and allow thus the actual image acquisition.

GVSP

As soon as the camera is configured through GVCP, it can start sending the image and other data. GigE Vision defines GVSP (GigE Vision Streaming Protocol) for this purpose. GVSP mainly defines, how the image data are packetized and it also offers means for recovery from possible errors occurring during the data transfer.

Bootstrap registers

The most basic camera parameters, allowing its identification and establishing the connection are summarized in a mandatory block of registers, so called bootstrap registers. The layout of additional registers defining specific camera functionality is left completely up to the camera implementation. The camera describes that layout using XML file compatible with the GenICam standard. This configuration file is usually stored directly on the camera and can be downloaded by means of GVCP.

9. Troubleshooting & Support

9.1. Getting support for Leutron Vision products

9.1.1. Before you contact the support team

To get the most effective support for your problems, please make sure that you

- Check if your issue does not belong to the list of the most common issues (providing solutions to all of them).
- Send the problem report to proper e-mail address, depending on your location and the problem nature, as explained in Section 9.1.2, “Contact options” [p. 70].
- Make sure the problem is well reproducible, preferably with the Simplon Explorer and that you capture the `lv.simplon.log` file (Section 9.1.3, “Gathering information about the problem” [p. 70]).
- Attach the captured log file (one per each session if multiple test scenarios are described in the report) as well as other important information, describe all details about the problem, see Section 9.1.4, “Reporting the problem” [p. 71].

Following these guidelines will guarantee that your problem can be solved effectively, in the shortest possible time.

9.1.2. Contact options

Following list should help to identify the e-mail contact to be used for your problem report:

- If you purchased Leutron Vision products through a distributor, please contact that distributor with every support case. The list of Leutron Vision distributors[1] with their contact information is provided in our website.
- If you reside in Germany or Austria, contact `<desupport@leutron.com>`.
- If you reside in US or Canada, contact `<ussupport@leutron.com>`.
- If you reside in Czech republic or Slovakia, contact `<czsupport@leutron.com>`.
- In all other cases contact
  - `<gigesupport@picsight.com>` for issues related to PicSight-GigE/PicSight-Smart cameras
  - `<linuxsupport@leutron.com>` for any issues with Simplon for Linux
  - and `<intsupport@leutron.com>` for any other products, issues and questions

9.1.3. Gathering information about the problem

When collecting information about your problem, please follow the steps listed hereafter:

1. **Be sure you use the latest version of Simplon.** It may be that you have an old version of Simplon installed and you are experiencing a bug, which was already removed in the meantime. The version of currently installed Simplon is visible in the Simplon Explorer tool (menu Help → About). The latest Simplon version can be downloaded from the Simplon

[1] http://www.leutron.com/sales-contact/

download area[2]. If you have not registered yet for Simplon download on our website, you will have to do so first.

2. **Enable logging.** Before attempting to reproduce the problem, be sure the logging is enabled through the Simplon Settings tool (or directly the `lv.simplon.ini` file (refer to Section 7.1, "Simplon Settings utility and the configuration file" [p. 65]). Enable logging to file, preferably with highest log level. If the application crashes and the log is not complete, switch on the `Disable all buffering` log configuration option.

3. **Test the hardware functionality with a supplied demo program, Simplon Explorer.** As the first step use the "standard" demo program, Simplon Explorer, which covers most of the possibilities of Leutron Vision hardware usage.

   If you are able to reproduce the problem in Simplon Explorer, then the problem is probably not in your code. However, it still need not be a bug, a problem can be also caused by an improper configuration.

4. **Camera configuration.** Check especially whether the camera is set in a proper working mode. See camera documentation to learn how the camera should be adjusted.

5. **Check with another hardware.** Sometimes the problem can be caused by the HW/SW environment or by a defective piece of hardware. If it is in your possibilities, try if the same problem happens on another PC. If you have another camera of the same type available, try if the problem happens also after the camera replacement. For GigE Vision based cameras, review the network configuration and try to connect the camera directly, without additional network components in the path.

6. **Check with your application.** If the problem could not be reproduced in the previous steps, reproduce the problem in your application. If it is possible, extract and send us the smallest part of your code that can demonstrate where the problem occurs. Even better is if you can make the extracted code fragment compilable, so that the problem can be reproduced with a small application that you can send to us. Make sure that your application does proper error handling when accessing Simplon functions. Ignoring the error status of Simplon functions can lead to unexpected behavior of your application.

7. **Screenshots.** In case of disturbances in the acquired images, make and send us the screenshots of them, either as bitmaps with lossless compression (`.bmp`, `.png`) or as `.jpg` with a high quality (small compression).

8. **Send us the report.** Send all the collected information (and especially the `lv.simplon.log` file) to us, following guidelines from Section 9.1.4, "Reporting the problem" [p. 71].

9.1.4. Reporting the problem

Each problem report should contain following information:

- The e-mail subject should give a brief but specific description of the problem and your company name (separated by a "@" character). The subject will be used to identify evolution of the support case, so if you need to submit a different problem, create a new subject string, do not reuse the old one.
  - Example of a good subject: *Changing exposure time has no effect @ My Company Inc.*
  - Example of a bad subject: *Problem with camera*
- With each new support case, please list your Simplon registration number in the e-mail.

[2] http://www.leutron.com/support-downloads/download-area/

- Describe your problem in detail, including all its symptoms, conditions under which it occurs and all other information that can help resolving the problem. List all the information you gathered while testing and reproducing the problem, as described in Section 9.1.3, “Gathering information about the problem” [p. 70].

- Attach the required `lv.simplon.log` file (one per each reported session). The file absolutely essential for our support team and failing to attach it will usually only lead to a response asking you to collect and send it, which will in effect make the time needed for the resolution longer. Be sure that you send the log file generated during the problematic session you are mentioning in your report reporting.

- If troubleshooting a CheckSight camera running Linux, attach also the device driver log (Section 9.2.3.1, “Linux driver logging” [p. 73]) and contents of `/proc/lvsm`. The driver logs are appended to the standard kernel log and can be retrieved for example using the `dmesg` command.

- If the problem occurs only under some conditions or only with some particular piece of hardware or software, describe the differences between the two cases and attach log files generated during both the successful and problematic sessions. If only a specific sequence of steps leads to the problem, describe this sequence.

9.1.5. Return Material Authorization (RMA)

If you need to return some (possibly defective) material to Leutron Vision, please use the RMA application[3] on our website. The application will query you for some important information about your product(s) and their defect(s), deciding automatically, whether you should send the material immediately back to Leutron Vision (or your distributor) or whether we will first contact you to do more tests for problem verification.

9.1.6. Customer registration

Customers willing to get (different level of) access rights to Leutron Vision download area[4] should first register on our website. The registration area[5] provides various registration possibilities from access to freely downloadable data (anyone can get access to it) to access to Simplon download[6] (you need to purchase Simplon first).

9.2. Troubleshooting Leutron Vision products

9.2.1. Troubleshooting common issues

The lists below mention some most common issues that can be faced by customers, but have easy solution. It is a good idea to check this list first when encountering a problem. It is also a good idea to check Section 9.2.2, “Known problems and limitations” [p. 73] and Section 9.2.3, “Simplon logging” [p. 73] whether some hints can be found there.

- If you are not getting any image, if you get different image data than expected or if you have problems with acquisition timing, check again carefully that the camera is set in proper working mode and that it is not in some error status (see camera manual for possible description of such situations).

- If you have problems with acquisition from a PicSight-GigE/PicSight-Smart camera, check whether your firewall settings do not interfere with the acquisition. PicSight-GigE user manual provides more information about this topic.

[3] http://www.leutron.com/rma/client_id.php
[4] http://www.leutron.com/support-downloads/download-area/
[5] http://www.leutron.com/support-downloads/registrations/
[6] http://www.leutron.com/support-downloads/download-area/

9.2.2. Known problems and limitations

(none in current version)

9.2.3. Simplon logging

All the Simplon components are using a common logging subsystem, which can be configured through the `lv.simplon.ini` and Simplon Settings (Section 7.1, “Simplon Settings utility and the configuration file” [p. 65]).

The log can be directed to various outputs, while the most important of them (especially for support) is the file output. The log file path can be adjusted, however the default name (`lv.simplon.log`) as well as the default location (application data directory under Windows, `/var/opt/simplon/log` under Linux) will be just right in most situations.

The logging level can also be adjusted. While the default level `Info (4)` provides most useful information about the system and Simplon operation, for specific troubleshooting and support requests, it is often useful to switch to the most verbose level `Trace (6).`

For the case of troubleshooting problems involving an application crash, when the log might not be written completely (log buffers not flushed completely at the time of application crash), the logging subsystem should be configured to avoid any buffering. In such case, please switch on the `Disable all buffering` option.

The logging configuration offers few more options, the most important of them being the max allowed size of the log file. The log can be also directed to other outputs, but those are mainly intended for use during advanced debugging by qualified personnel.

Several logging mechanisms are used by various Simplon components. They are separated to utilize log analysis, to separate domain-specific info from the general purpose logs and to improve performance, logging only what needs to get logged in a given situation. Creation of these log files can usually be controlled by the user.

Note however, that all the information in the logs is just informative and its main purpose is to assist the Leutron Vision support team when trying to resolve possible problems. It is not designed as a tool that should be necessarily well understandable by users, it might be misleading and all the information is context-specific. If a word “error” or “warning” appears in the log, it does not necessarily mean a real error occurred, it might be valid situation in a given context. You don't need to get alarmed because of that and contact support, especially if the application is otherwise working well.

9.2.3.1. Linux driver logging

On the CheckSight systems, a device driver controlling the acquisition device is running. The driver can also provide useful logs, but those are not routed through Simplon user level logging subsystem, but rather through standard Linux kernel logging mechanisms. Depending on the system configuration, there might be multiple ways to access that log, but the most convenient is usually to simply execture the `dmesg` command and store its output to a file.

Apart of the kernel log messages, other (mostly static) information can be available through reading the `/proc/lvsm`.

9.2.4. Debugging with GigE Vision cameras

When debugging an application using GigE Vision cameras (PicSight-GigE/PicSight-Smart), it's necessary to pay attention to a low level GigE Vision communication parameter, the heartbeat timeout.

This timeout, in milliseconds, defines how frequently has the camera send the “heartbeat” messages to the camera, indicating that it's still alive. When the camera does not receive the message within the specified timeout period, it assumes the application has gone away and disconnects itself. The default value used in Simplon is 3 s (other software packages might use another value).

At most times, the heartbeat handling is fully transparent to the user. However, during debugging, when a developer stops the application at a breakpoint for single stepping, the Simplon heartbeat handling thread would also stop, fail to deliver the heartbeat messages and the camera would disconnect. To prevent that, we recommend to increase the heartbeat timeout parameter to a higher value. The exact value should be a compromise between being able to debug comfortably, while understanding that when the application stops unexpectedly without disconnecting the camera properly, it will take a while before the camera disconnects automatically and becomes available for another connection.

The parameter is configurable through camera features at runtime (the feature's GenICam name is `GevHeartbeatTimeout`), while its default value can be adjusted through the Simplon Settings utility, configuration option `Heartbeat timeout`.

Note that some software packages attempt to increase the timeout automatically, whenever they find out that a debugger is attached to the process. We don't do that since we believe that the correct heartbeat timeout value can only be determined by the application developer, for reasons described above.

9.2.5. Debugging symbols

When debugging certain kinds of problems, such as application crash, it might be very useful if you can provide the support team with extended diagnostic information, such as the stack trace showing the crash point. The installer of Simplon for Windows allows to install “PDB files” — files containing the debugging information (symbols) for the Simplon libraries. We recommend to install these debug files on the development and testing systems, so that it is possible to generate the stack trace info whenever encountering a problem. The files are located in the `bin\Debug` subdirectory of the Simplon installation.

Note that in Linux version of Simplon the symbols are directly included in the libraries and executables.

Contacting Leutron Vision

Headquarters (Switzerland)

Address: Industriestrasse 57, CH-8152, Glattbrugg, Switzerland

Phone: ++41 44 809 88 22

Fax: ++41 44 809 88 29

E-mail (sales): <intsales@leutron.com>, e-mail (support): <intsupport@leutron.com>

Web: www.leutron.com[1]

Germany and Austria

Address: Macairestrasse 3, D-78467 Konstanz, Germany

Phone: ++49 7531 59 42 0

Fax: ++49 7531 59 42 99

E-mail (sales): <desales@leutron.com>, e-mail (support): <desupport@leutron.com>

Web: www.leutron.com/de/[2]

North America

Address: 38 East Street, Halifax, MA 02338, USA

Phone (corporate): ++1 888 442 22 69 x0, phone (sales): ++1 888 442 22 69 x1, phone (support): ++1 888 442 22 69 x2

Fax (corporate): ++1 781 353 39 29, fax (sales/support): ++1 781 353 39 43

E-mail (sales): <ussales@leutron.com>, e-mail (support): <ussupport@leutron.com>

Czech Republic

Address: Rokycanska 27, CZ-31200 Plzen, Czech Republic

Phone: ++420 377 260 342

Fax: ++420 377 260 944

E-mail (sales): <czsales@leutron.com>, e-mail (support): <czsupport@leutron.com>

Other countries

Customers residing in other countries should have a look at the list of our representatives[3] for a distributor in their country. If no distributor exists in their country, they should contact the headquarter office directly.

[1] http://www.leutron.com/
[2] http://www.leutron.com/de/
[3] http://www.leutron.com/sales-contact/

Copyright © Leutron Vision

Useful links

- To download software, documentation and other stuff, please visit our download area[4].
- Get more information about our support[5]. If you need to return some (defective) material to Leutron Vision, please visit the Return Material Authorization (RMA) page[6].
- Get information about prices and ordering[7].
- Find out detailed information about our hardware and software product range[8].

[4] http://www.leutron.com/support-downloads/download-area/
[5] http://www.leutron.com/support-downloads/
[6] http://www.leutron.com/rma/client_id.php
[7] http://www.leutron.com/price-request/
[8] http://www.leutron.com/products-machine-vision-image-acquisition/